# The True Sample Complexity of Active Learning*

Maria-Florina Balcan[1], Steve Hanneke[2], and Jennifer Wortman Vaughan[3]**

[1] School of Computer Science, Georgia Institute of Technology
ninamf@cc.gatech.edu
[2] Department of Statistics, Carnegie Mellon University
shanneke@stat.cmu.edu
[3] School of Engineering and Applied Sciences, Harvard University
jenn@seas.harvard.edu

**Abstract.** We describe and explore a new perspective on the sample complexity of active learning. In many situations where it was generally believed that active learning does not help, we show that active learning does help in the limit, often with exponential improvements in sample complexity. This contrasts with the traditional analysis of active learning problems such as non-homogeneous linear separators or depth-limited decision trees, in which $\Omega(1/\epsilon)$ lower bounds are common. Such lower bounds should be interpreted carefully; indeed, we prove that it is always possible to learn an $\epsilon$-good classifier with a number of samples asymptotically smaller than this. These new insights arise from a subtle variation on the traditional definition of sample complexity, not previously recognized in the active learning literature.

## 1 Introduction

Machine learning research has often focused on the problem of learning a classifier from a fixed set of labeled examples. However, for many contemporary practical problems such as classifying web pages or detecting spam, there is often an abundance of *unlabeled* examples available, from which only a relatively small subset may be labeled and used for learning. In such scenarios, the natural question that arises is how to best select a useful subset of examples to be labeled.

One possibility, which has recently generated substantial interest, is *active learning*. In active learning, the learning algorithm itself is allowed to select the subset of available examples to be labeled. The algorithm may request labels one at a time, using the requested label information from previously selected examples to inform its decision of which example to select next. The hope is that by only requesting the labels of "informative" examples, the algorithm can learn a good classifier using significantly fewer labels than would typically be required to learn a classifier from randomly chosen examples.

A number of active learning analyses have recently been proposed in a PAC-style setting, both for the realizable and for the agnostic cases, resulting in a sequence of important positive and negative results [2, 4, 8–12, 15, 17]. These include several general

* A preliminary version of this work appeared in the *Proceedings of the 21st Conference on Learning Theory*, 2008 [5].

** Most of this research was done while the author was at the University of Pennsylvania.

sample complexity bounds in terms of complexity parameters [10, 15, 16, 12, 17], thus giving general sufficient conditions for signficant improvements over passive learning. For instance, perhaps the most widely-studied concrete positive result for when active learning helps is that of learning homogeneous (i.e., through the origin) linear separators, when the data is linearly separable and distributed uniformly over the unit sphere [2, 4, 10–12]. However, in addition to these known positive results, there are simple (almost trivial) examples, such as learning intervals or non-homogeneous linear separators, where these analyses of sample complexity have indicated that perhaps active learning does not help at all [10, 16].

In this work, we approach the analysis of active learning algorithms from a different angle. Specifically, we point out that traditional analyses have studied the number of label requests required before an algorithm can both produce an $\epsilon$-good classifier *and* prove that the classifier's error is no more than $\epsilon$. These studies have turned up simple examples where this number is no smaller than the number of random labeled examples required for passive learning. This is the case for learning certain nonhomogeneous linear separators and intervals on the real line, and generally seems to be a common problem for many learning scenarios. As such, it has led some to conclude that active learning *does not help* for most learning problems. One of the goals of our present analysis is to dispel this misconception. Specifically, we study the number of labels an algorithm needs to request before it can produce an $\epsilon$-good classifier, even if there is no accessible confidence bound available to verify the quality of the classifier. With this type of analysis, we prove that active learning can essentially always achieve asymptotically superior sample complexity compared to passive learning when the VC dimension is finite. Furthermore, we find that for most natural learning problems, including the negative examples given in the previous literature, active learning can achieve exponential[4] improvements over passive learning with respect to dependence on $\epsilon$. This situation is characterized in Figure 1.1.

To our knowledge, this is the first work to address this subtle point in the context of active learning. Though several previous papers have studied bounds on this latter type of sample complexity [12, 11, 7], their results were *no stronger* than the results one could prove in the traditional analysis. As such, it seems this large gap between the two types of sample complexities has gone unnoticed until now.

### 1.1 A Simple Example: Intervals

To get some intuition about when these types of sample complexity are different, consider the following example. Suppose that $C$ is the class of all intervals over $[0, 1]$ and $D$ is a uniform distribution over $[0, 1]$. If the target function is the empty interval, then for any sufficiently small $\epsilon$, in order to *verify* with high confidence that this (or any) interval has error $\leq \epsilon$, we need to request labels in at least a constant fraction of the $\Omega(1/\epsilon)$ intervals $[0, 2\epsilon], [2\epsilon, 4\epsilon], \ldots$, requiring $\Omega(1/\epsilon)$ total label requests.

However, no matter what the target function is, we can *find* an $\epsilon$-good classifier with only a logarithmic sample complexity via the following extremely simple 2-phase

[4] We slightly abuse the term "exponential" throughout the paper. In particular, we refer to any $polylog(1/\epsilon)$ as being an exponential improvement over $1/\epsilon$.

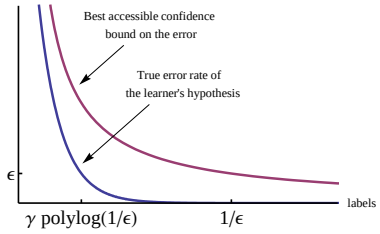


**Fig. 1.1.** Active learning can often achieve exponential improvements, though in many cases the amount of improvement cannot be detected from information available to the learning algorithm. Here $\gamma$ may be a target-dependent constant.

learning algorithm. The algorithm will be allowed to make $t$ label requests, and then we will find a value of $t$ that is sufficiently large to guarantee learning. We start with a large set of unlabeled examples. In the first phase, on each round we choose a point $x$ uniformly at random from the unlabeled sample and query its label. We repeat this until we either observe a $+1$ label, at which point we enter the second phase, or we use all $t$ label requests. In the second phase, we alternate between running one binary search on the examples between $0$ and that $x$ and a second on the examples between that $x$ and $1$ to approximate the end-points of the interval. Once we use all $t$ label requests, we output a smallest interval consistent with the observed labels.

If the target is an interval $[a, b] \subseteq [0, 1]$, where $b - a = w > 0$, then after roughly $O(1/w)$ queries (a constant number that depends only on the target), a positive example will be found. Since only $O(\log(1/\epsilon))$ additional queries are required to run the binary search to reach error rate $\epsilon$, it suffices to have $t \geq O(1/w + \log(1/\epsilon)) = O(\log(1/\epsilon))$. This essentially reflects the "two-phases" phenomenon noted by [10], where improvements are often observable only after some initial period, in this case the $1/w$ initial samples. On the other hand, if the target $h^*$ labels every point as $-1$ (the so-called *all-negative* function), the algorithm described above would output a hypothesis with $0$ error even after $0$ label requests, so any $t \geq 0$ suffices in this case. So in general, the sample complexity is at worst $O(\log(1/\epsilon))$. Thus, we see a sharp distinction between the sample complexity required to *find* a good classifier (logarithmic) and the sample complexity needed to both find a good classifier *and verify* that it is good.

This example is particularly simple, since there is effectively only *one* "hard" target function (the all-negative target). However, most of the spaces we study are significantly more complex than this, and there are generally many targets for which it is difficult to achieve good verifiable complexity.

### 1.2 Our Results

We show that in many situations where it was previously believed that active learning cannot help, active learning does help in the limit. Our main specific contributions are as follows:

– We distinguish between two different variations on the definition of sample complexity. The traditional definition, which we refer to as *verifiable sample complexity*, focuses on the number of label requests needed to obtain a confidence bound indicating an algorithm has achieved at most $\epsilon$ error. The newer definition, which we refer to simply as *sample complexity*, focuses on the number of label requests before an algorithm actually achieves at most $\epsilon$ error. We point out that the latter is often significantly smaller than the former, in contrast to passive learning where they are often equivalent up to constants for most nontrivial learning problems.
– We prove that *any* distribution and finite VC dimension concept class has active learning sample complexity asymptotically smaller than the sample complexity of passive learning for nontrivial targets. A simple corollary of this is that finite VC dimension implies $o(1/\epsilon)$ active learning sample complexity.
– We show it is possible to actively learn with an *exponential rate* a variety of concept classes and distributions, many of which are known to require a linear rate in the traditional analysis of active learning: for example, intervals on $[0, 1]$ and non-homogeneous linear separators under the uniform distribution.
– We show that even in this new perspective, there do exist lower bounds; it is possible to exhibit somewhat contrived distributions where exponential rates are not achievable even for some simple concept spaces (see Theorem 6). The learning problems for which these lower bounds hold are much more intricate than the lower bounds from the traditional analysis, and intuitively seem to represent the core of what makes a hard active learning problem.

## 2 Background and Notation

Let $\mathcal{X}$ be an instance space and $\mathcal{Y} = \{-1, 1\}$ be the set of possible labels. Let $C$ be the concept class, a set of measurable functions mapping from $\mathcal{X}$ to $\mathcal{Y}$, and assume that $C$ has VC dimension $d$. We consider here the realizable setting in which it is assumed that the instances are labeled by a target function $h^*$ in the class $C$. There is a distribution $D$ on $\mathcal{X}$, and the *error rate* of a hypothesis $h$ is defined as $\mathrm{er}(h) = \mathbb{P}_D(h(x) \neq h^*(x))$.

We assume the existence of an infinite sequence $x_1, x_2, \ldots$ of examples sampled i.i.d. according to $D$. The learning algorithm may access any finite prefix $x_1, x_2, \ldots, x_m$ of the sequence. Essentially, this means we allow the algorithm access to an arbitrarily large, but finite, sequence of random unlabeled examples. In active learning, the algorithm can select any example $x_i$, and request the label $h^*(x_i)$ that the target assigns to that example, observing the labels of all previous requests before selecting the next example to query. The goal is to find a hypothesis $h$ with small error with respect to $D$, while simultaneously minimizing the number of label requests that the learning algorithm makes.

### 2.1 Two Definitions of Sample Complexity

The following definitions present a subtle but significant distinction we refer to throughout the paper. Several of the results that follow highlight situations where these two definitions of sample complexity can have dramatically different dependence on $\epsilon$.

**Definition 1.** *A function $S(\epsilon, \delta, h^*)$ is a* verifiable sample complexity *for a pair $(C, D)$ if there exists an active learning algorithm $A(t, \delta)$ that outputs* both a classifier $h_{t,\delta}$ and a value $\hat{\epsilon}_{t,\delta} \in \mathbb{R}$ *after making at most $t$ label requests, such that for any target function $h^* \in C, \epsilon \in (0, 1/2), \delta \in (0, 1)$, for any $t \geq 0$, $\mathbb{P}_D(er(h_{t,\delta}) \leq \hat{\epsilon}_{t,\delta}) \geq 1 - \delta$ and for any $t \geq S(\epsilon, \delta, h^*)$,*

$$\mathbb{P}_D(er(h_{t,\delta}) \leq \hat{\epsilon}_{t,\delta} \leq \epsilon) \geq 1 - \delta.$$

**Definition 2.** *A function $S(\epsilon, \delta, h^*)$ is a* sample complexity *for a pair $(C, D)$ if there exists an active learning algorithm $A(t, \delta)$ that outputs a classifier $h_{t,\delta}$ after making at most $t$ label requests, such that for any target function $h^* \in C, \epsilon \in (0, 1/2), \delta \in (0, 1)$, for any $t \geq S(\epsilon, \delta, h^*)$,*

$$\mathbb{P}_D(er(h_{t,\delta}) \leq \epsilon) \geq 1 - \delta.$$

Let us take a moment to reflect on the difference between these two definitions, which may appear quite subtle. Both definitions allow the sample complexity to depend both on the target function and on the input distribution. The only distinction is whether or not there is an *accessible guarantee* or *confidence bound* on the error of the chosen hypothesis that is also at most $\epsilon$. This confidence bound can only depend on quantities accessible to the learning algorithm, such as the $t$ requested labels. As an illustration of this distinction, consider again the problem of learning intervals. As described above, there exists an active learning algorithm such that, given a large enough initial segment of the unlabeled data, if the target $h^*$ is an interval of width $w$, then after seeing $O(1/w + \log(1/\epsilon))$ labels, with high probability the algorithm outputs a classifier with error rate less than $\epsilon$ *and* a guarantee that the error rate is less than $\epsilon$. In this case, for sufficiently small $\epsilon$, the verifiable sample complexity $S(\epsilon, \delta, h^*)$ is proportional to $\log(1/\epsilon)$. However, if $h^*$ is the all-negative function, then the verifiable sample complexity is at least proportional to $1/\epsilon$ for *all* values of $\epsilon$ because *a high-confidence guarantee can never be made* without observing $\Omega(1/\epsilon)$ labels; for completeness, a formal proof of this fact is included in Appendix A. In contrast, as we have seen, there is an algorithm that, given a large enough initial segment of the unlabeled sequence, produces a classifier with error rate less than $\epsilon$ after a number of label requests $O(\log(1/\epsilon))$ for *every* target in the class of intervals; thus, it is possible to acheive sample complexity $O(\log(1/\epsilon))$ for *every* target in the class of intervals.

Any verifiable sample complexity function is also a sample complexity function, but we study a variety of cases where the reverse is not true. In situations where there are sample complexity functions significantly smaller than any achievable verifiable sample complexities, we sometimes refer to the smaller quantity as the *true sample complexity* to distinguish it from the verifiable sample complexity.

A common alternative formulation of verifiable sample complexity is to let $A$ take $\epsilon$ as an argument and allow it to choose online how many label requests it needs in order to guarantee error at most $\epsilon$ [10, 2, 15, 16, 4]. This alternative definition is almost equivalent, as the algorithm must be able to produce a confidence bound of size at most $\epsilon$ on the error of its hypothesis in order to decide when to stop requesting labels anyway. In particular, any algorithm for either definition can be modified to fit the other definition without significant loss in the verifiable sample complexity values. For instance, given any algorithm for the alternative formulation, and given a value of $t$, we

can simply run the algorithm with argument $2^{-i}$ for increasing values of $i$ (and confidence parameter $\delta/(2i^2)$) until we use $t$ queries, and then take the output of the last run that was completed; for a reasonable algorithm, the sample complexity should increase as $\epsilon$ decreases, and we typically expect logarithmic dependence on the confidence parameter, so the increase in sample complexity due to these extra runs is at most a factor of $\tilde{O}(\log(1/\epsilon))$. Similarly, given any algorithm for Definition 1, and given $\epsilon$ as input, we might simply double $t$ until $\hat{\epsilon}_{t,\delta/(2t^2)} \leq \epsilon$, giving an algorithm for the alternative formulation; again, for reasonable algorithms, the sample complexity of the converted algorithm will be at most a factor of $O(\log(1/\epsilon))$ larger than the original.

Generally, there is some question as to what the "right" formal model of active learning is. For instance, we could instead let $A$ generate an infinite sequence of $h_t$ hypotheses (or $(h_t, \hat{\epsilon}_t)$ in the verifiable case), where $h_t$ can depend only on the first $t$ label requests made by the algorithm along with some initial segment of unlabeled examples (as in [7]), representing the case where we are not sure a priori of when we will stop the algorithm. However, for our present purposes, this alternative too is almost equivalent in sample complexity.

### 2.2 The Verifiable Sample Complexity

To date, there has been a significant amount of work studying the verifiable sample complexity (though typically under the aforementioned alternative formulation). It is clear from standard results in passive learning that verifiable sample complexities of $O\left((d/\epsilon)\log(1/\epsilon) + (1/\epsilon)\log(1/\delta)\right)$ are easy to obtain for any learning problem, by requesting the labels of random examples. As such, there has been much interest in determining when it is possible to achieve verifiable sample complexity *smaller* than this, and in particular, when the verifiable sample complexity is a polylogarithmic function of $1/\epsilon$ (representing exponential improvements over passive learning).

One of the earliest active learning algorithms in this model is the selective sampling algorithm of Cohn, Atlas, and Ladner [8], henceforth referred to as CAL. This algorithm keeps track of two spaces—the current *version space* $C_i$, defined as the set of hypotheses in $C$ consistent with all labels revealed so far, and the current *region of uncertainty* $R_i = \{x \in \mathcal{X} : \exists h_1, h_2 \in C_i \text{ s.t. } h_1(x) \neq h_2(x)\}$. In each round $i$, the algorithm picks a random unlabeled example from $R_i$ and requests its label, eliminating all hypotheses in $C_i$ inconsistent with the received label to make the next version space $C_{i+1}$. The algorithm then defines $R_{i+1}$ as the region of uncertainty for the new version space $C_{i+1}$ and continues. Its final hypothesis can then be taken arbitrarily from $C_t$, the final version space, and we use the diameter of $C_t$ for the $\hat{\epsilon}_t$ error bound. While there are a small number of cases in which this algorithm and others have been shown to achieve exponential improvements in the verifiable sample complexity for all targets (most notably, the case of homogeneous linear separators under the uniform distribution), there exist extremely simple concept classes for which $\Omega(1/\epsilon)$ labels are needed for some targets.

Recently, there have been a few quantities proposed to measure the verifiable sample complexity of active learning on any given concept class and distribution. Dasgupta's *splitting index* [10], which is dependent on the concept class, data distribution, and target function, quantifies how easy it is to make progress toward reducing the diameter

of the version space by choosing an example to query. Another quantity to which we will frequently refer is Hanneke's *disagreement coefficient* [15], defined as follows.

**Definition 3.** *For any set of classifiers $H$, define the* region of disagreement *of $H$ as*

$$\mathrm{DIS}(H) = \{x \in \mathcal{X} : \exists h_1, h_2 \in H : h_1(x) \neq h_2(x)\} .$$

*For any classifier $h$ and $r > 0$, let $\tilde{B}(h, r)$ be a ball of radius $r$ around $h$ in $C$. Formally,*

$$\tilde{B}(h, r) = \{h' \in \tilde{C} : \mathbb{P}_D(h(x) \neq h'(x)) \leq r\} ,$$

*where $\tilde{C}$ denotes any countable dense subset of $C$.*[5] *For our purposes, the* disagreement coefficient of a hypothesis $h$*, denoted $\theta_h$, is defined as*

$$\theta_h = \sup_{r>0} \frac{\mathbb{P}(\mathrm{DIS}(\tilde{B}(h, r)))}{r}.$$

*The disagreement coefficient for a* concept class $C$ *is defined as $\theta = \sup_{h \in C} \theta_h$.*

The disagreement coefficient is often a useful quantity for analyzing the verifiable sample complexity of active learning algorithms. For example, it has been shown that the algorithm of Cohn, Atlas, and Ladner described above achieves a verifiable sample complexity at most $\theta_{h^*} d \cdot \mathrm{polylog}(1/(\epsilon\delta))$ when run with hypothesis class $\tilde{C}$ for target function $h^* \in C$ [15, 17]. We will use it in several of the results below.

To get a feel for how to calculate this quantity, it may be helpful to see some examples (taken from [15]). For instance, consider $D$ uniform on $[0, 1]$, and the concept space of threshold classifiers $C = \{h_z : z \in [0, 1], h_z(x) = +1 \text{ iff } x \geq z\}$. In this case, we have $\tilde{B}(h_z, r) \subseteq \{h_{z'} : |z' - z| \leq r\}$, so $\mathrm{DIS}(\tilde{B}(h_z, r)) \subseteq \{x : |x - z| \leq r\}$, and thus $\mathbb{P}(\mathrm{DIS}(\tilde{B}(h_z, r))) \leq 2r$. Therefore, the disagreement coefficient of $h_z$ is $\leq 2$, and in fact so is the disagreement coefficient for the entire concept class.

On the other hand, consider the same $D$, but this time take the concept class of intervals: $C = \{h_{a,b} : a, b \in [0, 1], h_{a,b}(x) = +1 \text{ iff } a \leq x \leq b\}$. In this case, for $h_{a,b}$ with $|a - b| = w > 0$, we have two cases. If $r > w$, $\{h_{a',b'} \in \tilde{C} : |a' - b'| \leq r - w\} \subseteq \tilde{B}(h_{a,b}, r)$, so that $\mathbb{P}(\mathrm{DIS}(\tilde{B}(h_{a,b}, r))) = 1$. In the second case, if $r < w$ we have $\tilde{B}(h_{a,b}, r) \subseteq \{h_{a',b'} : |a - a'| \leq r, |b - b'| \leq r\}$, so that $\mathrm{DIS}(\tilde{B}(h_{a,b}, r)) \subseteq \{x : \min\{|x - a|, |x - b|\} \leq r\}$, and thus $\mathbb{P}(\mathrm{DIS}(\tilde{B}(h_{a,b}, r))) \leq 4r$. Combining the two cases, we have $1/w \leq \theta_{h_{a,b}} \leq \max\{1/w, 4\}$. However, for the intervals with $|a - b| = 0$, the first case holds for arbitrarily small $r$ values, implying $\theta_{h_{a,b}} = \infty$.

We will see that both the disagreement coefficient and splitting index are also useful quantities for analyzing true sample complexities, though their use in that case is less direct.

[5] That is, $\tilde{C}$ is countable and $\forall h \in C, \forall \epsilon > 0, \exists h' \in \tilde{C} : \mathbb{P}(h(X) \neq h'(X)) \leq \epsilon$. Such a subset exists, for example, in any $C$ with finite VC dimension. We introduce this countable dense subset to avoid certain degenerate behaviors, such as when $\mathrm{DIS}(B(h, 0)) = \mathcal{X}$. For instance the hypothesis class of classifiers on the $[0, 1]$ interval that label exactly one point positive has this property under any density function.

### 2.3 The True Sample Complexity

This paper focuses on situations where true sample complexities are significantly smaller than verifiable sample complexities. In particular, we show that many common pairs $(C, D)$ have sample complexity that is polylogarithmic in *both* $1/\epsilon$ and $1/\delta$ and linear only in some finite target-dependent constant $\gamma_{h^*}$. This contrasts sharply with the infamous $1/\epsilon$ lower bounds mentioned above, which have been identified for verifiable sample complexity [16, 10, 9, 14]. The implication is that, for any fixed target $h^*$, such lower bounds vanish as $\epsilon$ approaches 0. This also contrasts with passive learning, where $1/\epsilon$ lower bounds are typically unavoidable [1].

**Definition 4.** *We say that* $(C, D)$ *is* actively learnable at an exponential rate *if there exists an active learning algorithm achieving sample complexity*

$$S(\epsilon, \delta, h^*) = \gamma_{h^*} \cdot \text{polylog}\left(1/(\epsilon\delta)\right)$$

*for all* $h^* \in C$*, where* $\gamma_{h^*}$ *is a finite constant that may depend on* $h^*$ *and* $D$ *but is independent of* $\epsilon$ *and* $\delta$*.*

## 3 Strict Improvements of Active Over Passive

In this section, we describe conditions under which active learning can achieve a sample complexity asymptotically superior to passive learning. The results are surprisingly general, indicating that whenever the VC dimension is finite, *any* passive learning algorithm is asymptotically *dominated* by an active learning algorithm on *all* targets.

**Definition 5.** *A function* $S(\epsilon, \delta, h^*)$ *is a* passive learning *sample complexity for a pair* $(C, D)$ *if there exists an algorithm* $A(((x_1, h^*(x_1)), (x_2, h^*(x_2)), \ldots, (x_t, h^*(x_t))), \delta)$ *that outputs a classifier* $h_{t,\delta}$*, such that for any target function* $h^* \in C, \epsilon \in (0, 1/2), \delta \in (0, 1)$*, for any* $t \geq S(\epsilon, \delta, h^*)$*,* $\mathbb{P}_D(er(h_{t,\delta}) \leq \epsilon) \geq 1 - \delta$*.*

Thus, a passive learning sample complexity corresponds to a restriction of an active learning sample complexity to algorithms that specifically request the first $t$ labels in the sequence and ignore the rest. In particular, it is known that for any finite VC dimension class, there is always an $O\left(1/\epsilon\right)$ passive learning sample complexity [18]. Furthermore, this is often (though not always) tight, in the sense that for any passive algorithm, there exist targets for which the corresponding passive learning sample complexity is $\Omega\left(1/\epsilon\right)$ [1]. The following theorem states that for any passive learning sample complexity, there exists an achievable active learning sample complexity with a strictly slower asymptotic rate of growth. Its proof is included in Appendix E.

**Theorem 1.** *Suppose* $C$ *has finite VC dimension, and let* $D$ *be any distribution on* $\mathcal{X}$*. For any passive learning sample complexity* $S_p(\epsilon, \delta, h)$ *for* $(C, D)$*, there exists an active learning algorithm achieving a sample complexity* $S_a(\epsilon, \delta, h)$ *such that, for all* $\delta \in (0, 1/4)$ *and targets* $h^* \in C$ *for which* $S_p(\epsilon, \delta, h^*) = \omega(1)$*,*[6]

$$S_a(\epsilon, \delta, h^*) = o\left(S_p(\epsilon/4, \delta, h^*)\right).$$

[6] Recall that we say a non-negative function $\phi(\epsilon) = o\left(1/\epsilon\right)$ iff $\lim_{\epsilon \to 0} \phi(\epsilon)/(1/\epsilon) = 0$. Similarly, $\phi(\epsilon) = \omega(1)$ iff $\lim_{\epsilon \to 0} 1/\phi(\epsilon) = 0$. Here and below, the $o(\cdot)$, $\omega(\cdot)$, $\Omega(\cdot)$ and $O(\cdot)$ notation

In particular, this implies the following simple corollary.

**Corollary 1.** *For any $C$ with finite VC dimension, and any distribution $D$ over $\mathcal{X}$, there is an active learning algorithm that achieves a sample complexity $S(\epsilon, \delta, h^*)$ such that for $\delta \in (0, 1/4)$,*

$$S(\epsilon, \delta, h^*) = o\left(1/\epsilon\right)$$

*for all targets $h^* \in C$.*

*Proof.* Let $d$ be the VC dimension of $C$. The passive learning algorithm of Haussler, Littlestone & Warmuth [18] is known to achieve a sample complexity no more than $(kd/\epsilon)\log(1/\delta)$, for some universal constant $k < 200$. Applying Theorem 1 now implies the result. □

Note the interesting contrast, not only to passive learning, but also to the known results on the *verifiable* sample complexity of active learning. This theorem definitively states that the $\Omega\left(1/\epsilon\right)$ lower bounds common in the literature on verifiable samples complexity can *never* arise in the analysis of the true sample complexity of finite VC dimension classes.

## 4 Decomposing Hypothesis Classes

Let us return once more to the simple example of learning the class of intervals over $[0, 1]$ under the uniform distribution. As discussed above, it is well known that the verifiable sample complexity of the all-negative classifier in this class is $\Omega(1/\epsilon)$. However, consider the more limited class $C' \subset C$ containing only the intervals $h$ of width $w_h$ strictly greater than 0. Using the simple algorithm described in Section 1.1, this restricted class can be learned with a (verifiable) sample complexity of only $O(1/w_h + \log(1/\epsilon))$. Furthermore, the remaining set of classifiers $C'' = C \setminus C'$ consists of only a single function (the all-negative classifier) and thus can be learned with verifiable sample complexity $0$. Here we have that $C$ can be decomposed into two subclasses $C'$ and $C''$, where both $(C', D)$ and $(C'', D)$ are learnable at an exponential rate. It is natural to wonder if the existence of such a decomposition is enough to imply that $C$ itself is learnable at an exponential rate.

More generally, suppose that we are given a distribution $D$ and a hypothesis class $C$ such that we can construct a sequence of subclasses $C_i$ with sample complexity $S_i(\epsilon, \delta, h)$, with $C = \cup_{i=1}^{\infty} C_i$. Thus, if we knew *a priori* that the target $h^*$ was a member of subclass $C_i$, it would be straightforward to achieve $S_i(\epsilon, \delta, h^*)$ sample complexity. It turns out that it is possible to learn *any* target $h^*$ in *any* class $C_i$ with sample complexity only $O(S_i(\epsilon/2, \delta/2, h^*))$, even without knowing which subclass the target belongs to in advance. This can be accomplished by using a simple aggregation

should be interpreted as $\epsilon \to 0$ (from the $+$ direction), treating all other parameters (e.g., $\delta$ and $h^*$) as fixed constants. Note that any algorithm achieving a sample complexity $S_p(\epsilon, \delta, h) \neq \omega(1)$ is guaranteed, with probability $\geq 1 - \delta$, to achieve error zero using a finite number of samples, and therefore we cannot hope to achieve a slower asymptotic growth in sample complexity.

algorithm, such as the one given below. Here a set of active learning algorithms (for example, multiple instances of Dasgupta's splitting algorithm [10] or CAL) are run on individual subclasses $C_i$ in parallel. The output of one of these algorithms is selected according to a sequence of comparisons. Specifically, for each pair of hypotheses, say $h_i$ and $h_j$ (where $h_i$ is produced by running the algorithm for $C_i$ and $h_j$ is produced by running the algorithm for $C_j$), we find a number of points $x$ on which $h_i(x) \neq h_j(x)$, and request their labels. With only a few such labels, we get quite a bit of information about whether $er(h_i) < er(h_j)$ or vice versa. We then select the $h_i$ making the smallest number of mistakes in the worst of these comparisons, and can conclude that its error rate cannot be much worse than any other $h_j$.

**Algorithm 1** The Aggregation Procedure. Here it is assumed that $C = \cup_{i=1}^{\infty} C_i$, and that for each $i$, $A_i$ is an algorithm achieving sample complexity at most $S_i(\epsilon, \delta, h)$ for the pair $(C_i, D)$. Both the main aggregation procedure and each algorithm $A_i$ take a number of labels $t$ and a confidence parameter $\delta$ as parameters.

Let $k$ be the largest integer s.t. $k^2 \lceil 72 \ln(4k/\delta) \rceil \leq t/2$
**for** $i = 1, \ldots, k$ **do**
  Let $h_i$ be the output of running $A_i(\lfloor t/(4i^2) \rfloor, \delta/2)$ on the sequence $\{x_{2n-1}\}_{n=1}^{\infty}$
**end for**
**for** $i, j \in \{1, 2, \ldots, k\}$ **do**
  **if** $\mathbb{P}_D(h_i(x) \neq h_j(x)) > 0$ **then**
    Let $R_{ij}$ be the first $\lceil 72 \ln(4k/\delta) \rceil$ elements $x$ in $\{x_{2n}\}_{n=1}^{\infty}$ with $h_i(x) \neq h_j(x)$
    Request the labels of all examples in $R_{ij}$
    Let $m_{ij}$ be the number of elements in $R_{ij}$ on which $h_i$ makes a mistake
  **else**
    Let $m_{ij} = 0$
  **end if**
**end for**
Return $\hat{h}_t = h_i$ where $i = \operatorname{argmin}_{i \in \{1,2,\ldots,k\}} \max_{j \in \{1,2,\ldots,k\}} m_{ij}$

Using this algorithm, we can show the following sample complexity bound. The proof appears in Appendix B.

**Theorem 2.** *For any distribution $D$, let $C_1, C_2, \ldots$ be a sequence of classes such that for each $i$, the pair $(C_i, D)$ has sample complexity at most $S_i(\epsilon, \delta, h)$ for all $h \in C_i$. Let $C = \cup_{i=1}^{\infty} C_i$. Then $(C, D)$ has a sample complexity at most*

$$\min_{i: h \in C_i} \max \left\{ 4i^2 \left\lceil S_i(\epsilon/2, \delta/2, h) \right\rceil, 2i^2 \left\lceil 72 \ln \frac{4i}{\delta} \right\rceil \right\},$$

*for any $h \in C$. In particular, Algorithm 1 achieves this when given as input the algorithms $A_i$ that each achieve sample complexity $S_i(\epsilon, \delta, h)$ on class $(C_i, D)$.*

A particularly interesting implication of Theorem 2 is that the ability to decompose $C$ into a sequence of classes $C_i$ with each pair $(C_i, D)$ learnable at an exponential rate is

enough to imply that $(C, D)$ is also learnable at an exponential rate. Since the *verifiable* sample complexity of active learning has received more attention and is therefore better understood, it is often useful to apply this result when there exist known bounds on the verifiable sample complexity; the approach loses nothing in generality, as suggested by the following theorem. The proof of this theorem. is included in Appendix C.

**Theorem 3.** *For any $(C, D)$ learnable at an exponential rate, there exists a sequence $C_1, C_2, \ldots$ with $C = \cup_{i=1}^{\infty} C_i$, and a sequence of active learning algorithms $A_1, A_2, \ldots$ such that the algorithm $A_i$ achieves* verifiable *sample complexity at most $\gamma_i polylog_i\left(1/(\epsilon\delta)\right)$ for the pair $(C_i, D)$, where $\gamma_i$ is a constant independent of $\epsilon$ and $\delta$. In particular, the aggregation algorithm (Algorithm 1) achieves exponential rates when used with these algorithms.*

Note that decomposing a given $C$ into a sequence of $C_i$ subsets that have good verifiable sample complexities is not always a simple task. One might be tempted to think a simple decomposition based on increasing values of verifiable sample complexity with respect to $(C, D)$ would be sufficient. However, this is not always the case, and generally we need to use information more detailed than verifiable complexity with respect to $(C, D)$ to construct a good decomposition. We have included in Appendix D a simple heuristic approach that can be quite effective, and in particular yields good sample complexities for every $(C, D)$ described in Section 5.

Since it is more abstract and allows us to use known active learning algorithms as a black box, we frequently rely on the decompositional view introduced here throughout the remainder of the paper.

## 5 Exponential Rates

The results in Section 3 tell us that the sample complexity of active learning can be made strictly superior to any passive learning sample complexity when the VC dimension is finite. We now ask how much better that sample complexity can be. In particular, we describe a number of concept classes and distributions that are learnable at an *exponential* rate, many of which are known to require $\Omega(1/\epsilon)$ *verifiable* sample complexity.

### 5.1 Exponential rates for simple classes

We begin with a few simple observations, to point out situations in which exponential rates are trivially achievable; in fact, in each of the cases mentioned in this subsection, the sample complexity is actually $O(1)$.

Clearly if $|\mathcal{X}| < \infty$ or $|C| < \infty$, we can always achieve exponential rates. In the former case, we may simply request the label of every $x$ in the support of $D$, and thereby perfectly identify the target. The corresponding $\gamma = |\mathcal{X}|$. In the latter case, the CAL algorithm can achieve exponential learning with $\gamma = |C|$ since each queried label will reduce the size of the version space by at least one.

Less obvious is the fact that a similar argument can be applied to any *countably infinite* hypothesis class $C$. In this case we can impose an ordering $h_1, h_2, \cdots$ over the classifiers in $C$, and set $C_i = \{h_i\}$ for all $i$. By Theorem 2, applying the aggregation

procedure to this sequence yields an algorithm with sample complexity $S(\epsilon, \delta, h_i) = 2i^2 \lceil 72 \ln(4i/\delta) \rceil = O(1)$.

### 5.2 Geometric Concepts, Uniform Distribution

Many interesting geometric concepts in $\mathbb{R}^n$ are learnable at an exponential rate if the underlying distribution is uniform on some subset of $\mathbb{R}^n$. Here we provide some examples; interestingly, every example in this subsection has some targets for which the *verifiable* sample complexity is $\Omega(1/\epsilon)$. As we see in Section 5.3, all of the results in this section can be extended to many other types of distributions as well.

**Unions of $k$ intervals under arbitrary distributions:** Let $\mathcal{X}$ be the interval $[0, 1)$ and let $C^{(k)}$ denote the class of unions of at most $k$ intervals. In other words, $C^{(k)}$ contains functions described by a sequence $\langle a_0, a_1, \cdots, a_\ell \rangle$, where $a_0 = 0$, $a_\ell = 1$, $\ell \leq 2k + 1$, and $a_0, \cdots, a_\ell$ is the (nondecreasing) sequence of transition points between negative and positive segments (so $x$ is labeled $+1$ iff $x \in [a_i, a_{i+1})$ for some *odd* $i$). For any distribution, this class is learnable at an exponential rate by the following decomposition argument. First, define $C_1$ to be the set containing the all-negative function along with any functions that are equivalent given the distribution $D$. Formally,

$$C_1 = \{h \in C^{(k)} : \mathbb{P}(h(X) = +1) = 0\} .$$

Clearly $C_1$ has verifiable sample complexity 0. For $i = 2, 3, \ldots, k + 1$, let $C_i$ be the set containing all functions that can be represented as unions of $i - 1$ intervals but cannot be represented as unions of fewer intervals. More formally, we can inductively define each $C_i$ as

$$C_i = \left\{h \in C^{(k)} : \exists h' \in C^{(i-1)} \text{ s.t. } \mathbb{P}(h(X) \neq h'(X)) = 0\right\} \setminus \cup_{j<i} C_j .$$

For $i > 1$, within each subclass $C_i$, for each $h \in C_i$ the disagreement coefficient is bounded by something proportional to $k + 1/w(h)$, where $w(h)$ is the weight of the smallest positive or negative interval. Note $w(h) > 0$ by construction of the $C_i$ sets. Thus running CAL with $\tilde{C}_i$ achieves polylogarithmic (verifiable) sample complexity for any $h \in C_i$. Since $C^{(k)} = \cup_{i=1}^{k+1} C_i$, by Theorem 2, $C^{(k)}$ is learnable at an exponential rate.

**Ordinary Binary Classification Trees:** Let $\mathcal{X}$ be the cube $[0, 1]^n$, $D$ be the uniform distribution on $\mathcal{X}$, and $C$ be the class of binary decision trees using a finite number of axis-parallel splits (see e.g., Devroye et al. [13], Chapter 20). In this case, in the same spirit as the previous example, we let $C_i$ be the set of decision trees in $C$ distance zero from a tree with $i$ leaf nodes, not contained in any $C_j$ for $j < i$. For any $i$, the disagreement coefficient for any $h \in C_i$ (with respect to $(C_i, D)$) is a finite constant, and we can choose $\tilde{C}_i$ to have finite VC dimension, so each $(C_i, D)$ is learnable at an exponential rate (by running CAL with $\tilde{C}_i$). By Theorem 2, $(C, D)$ is learnable at an exponential rate.

**Linear Separators**

**Theorem 4.** *Let $C$ be the concept class of linear separators in $n$ dimensions, and let $D$ be the uniform distribution over the surface of the unit sphere. The pair $(C, D)$ is learnable at an exponential rate.*

*Proof.* There are multiple ways to achieve this. We describe here a simple proof that uses a decomposition as follows. Let $\lambda(h)$ be the probability mass of the minority class under hypothesis $h$. Let $C_1$ be the set containing only the separators $h$ with $\lambda(h) = 0$, let $C_2 = \{h \in C : \lambda(h) = 1/2\}$, and let $C_3 = C \backslash (C_1 \cup C_2)$. As before, we can use a black box active learning algorithm such as CAL to learn within the class $C_3$. To prove that we indeed get the desired exponential rate of active learning, we show that the disagreement coefficient of any separator $h \in C_3$ with respect to $(C_3, D)$ is finite. Hanneke's results concerning the CAL algorithm [15, 17] then immediately imply that $C_3$ is learnable at an exponential rate. Since $C_1$ trivially has sample complexity 1, and $(C_2, D)$ is known to be learnable at an exponential rate [10, 4, 15, 12], combined with Theorem 2, this would imply the result. Below, we will restrict the discussion to hypotheses in $C_3$, which will be implicit in notation such as $B(h, r)$, etc.

First note that, to show $\theta_h < \infty$, it suffices to show that

$$\lim_{r \to 0} \frac{\mathbb{P}(DIS(B(h, r)))}{r} < \infty, \tag{5.1}$$

so we will focus on this.

For any $h$, there exists $r_h > 0$ s.t. $\forall h' \in B(h, r), \mathbb{P}(h'(X) = +1) \leq 1/2 \Leftrightarrow \mathbb{P}(h(X) = +1) \leq 1/2$, or in other words the minority class is the same among all $h' \in B(h, r)$. Now consider any $h' \in B(h, r)$ for $0 < r < \min\{r_h, \lambda(h)/2\}$. Clearly $\mathbb{P}(h(X) \neq h'(X)) \geq |\lambda(h) - \lambda(h')|$. Suppose $h(x) = sign(w \cdot x + b)$ and $h'(x) = sign(w' \cdot x + b')$ (where, without loss, we assume $\|w\| = 1$), and $\alpha(h, h') \in [0, \pi]$ is the angle between $w$ and $w'$. If $\alpha(h, h') = 0$ or if the minority regions of $h$ and $h'$ do not intersect, then clearly $\mathbb{P}(h(X) \neq h'(X)) \geq \frac{2\alpha(h,h')}{\pi} \min\{\lambda(h), \lambda(h')\}$. Otherwise, consider the classifiers $\bar{h}(x) = sign(w \cdot x + \bar{b})$ and $\bar{h}'(x) = sign(w' \cdot x + \bar{b}')$, where $\bar{b}$ and $\bar{b}'$ are chosen s.t. $\mathbb{P}(\bar{h}(X) = +1) = \mathbb{P}(\bar{h}'(X) = +1)$ and $\lambda(\bar{h}) = \min\{\lambda(h), \lambda(h')\}$. That is, $\bar{h}$ and $\bar{h}'$ are identical to $h$ and $h'$ except that we adjust the bias term of the one with larger minority class probability to reduce its minority class probability to be equal to the other's. If $h \neq \bar{h}$, then most of the probability mass of $\{x : h(x) \neq \bar{h}(x)\}$ is contained in the majority class region of $h'$ (or vice versa if $h' \neq \bar{h}'$), and in fact every point in $\{x : h(x) \neq \bar{h}(x)\}$ is labeled by $\bar{h}$ according to the majority class label (and similarly for $h'$ and $\bar{h}'$). Therefore, we have $\mathbb{P}(h(X) \neq h'(X)) \geq \mathbb{P}(\bar{h}(X) \neq \bar{h}'(X))$.

We also have that $\mathbb{P}(\bar{h}(X) \neq \bar{h}'(X)) \geq \frac{2\alpha(h,h')}{\pi} \lambda(\bar{h})$. To see this, consider the projection onto the 2-dimensional plane defined by $w$ and $w'$, as in Figure 5.2. Because the two decision boundaries must intersect inside the acute angle, the probability mass contained in each of the two wedges (both with $\alpha(h, h')$ angle) making up the projected region of disagreement between $\bar{h}$ and $\bar{h}'$ must be at least an $\alpha(h, h')/\pi$ fraction of the total minority class probability for the respective classifier, implying the union of these two wedges has probability mass at least $\frac{2\alpha(h,h')}{\pi} \lambda(\bar{h})$. Therefore, we must have

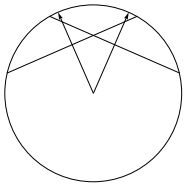

**Fig. 5.1.** Projection of $\bar{h}$ and $\bar{h}'$ into the plane defined by $w$ and $w'$.

$\mathbb{P}(h(X) \neq h'(X)) \geq \max\left\{|\lambda(h) - \lambda(h')|, \frac{2\alpha(h,h')}{\pi}\min\{\lambda(h), \lambda(h')\}\right\}$, and thus

$$B(h,r) \subseteq \left\{h' : \max\left\{|\lambda(h) - \lambda(h')|, \frac{2\alpha(h,h')}{\pi}\min\{\lambda(h), \lambda(h')\}\right\} \leq r\right\}.$$

The region of disagreement of this set is at most

$$\begin{aligned} DIS\left(\left\{h' : \frac{2\alpha(h,h')}{\pi}(\lambda(h) - r) \leq r \wedge |\lambda(h) - \lambda(h')| \leq r\right\}\right) \\ \subseteq DIS(\{h' : w' = w \wedge |\lambda(h') - \lambda(h)| \leq r\}) \\ \cup DIS(\{h' : \alpha(h,h') \leq \pi r/\lambda(h) \wedge |\lambda(h) - \lambda(h')| = r\}), \end{aligned}$$

where this last relation follows from the following reasoning. Take $y_{maj}$ to be the majority class of $h$ (arbitrary if $\lambda(h) = 1/2$). For any $h'$ with $|\lambda(h) - \lambda(h')| < r$, the $h''$ with $\alpha(h,h'') = \alpha(h,h')$ having $\mathbb{P}(h(X) = y_{maj}) - \mathbb{P}(h''(X) = y_{maj}) = r$ disagrees with $h$ on a set of points containing $\{x : h'(x) \neq h(x) = y_{maj}\}$; likewise, the one having $\mathbb{P}(h(X) = y_{maj}) - \mathbb{P}(h''(X) = y_{maj}) = -r$ disagrees with $h$ on a set of points containing $\{x : h'(x) \neq h(x) = -y_{maj}\}$. So any point in disagreement between $h$ and some $h'$ with $|\lambda(h) - \lambda(h')| < r$ and $\alpha(h,h') \leq \pi r/\lambda(h)$ is also disagreed upon by some $h''$ with $|\lambda(h) - \lambda(h'')| = r$ and $\alpha(h,h'') \leq \pi r/\lambda(h)$.

Some simple trigonometry shows that $DIS(\{h' : \alpha(h,h') \leq \pi r/\lambda(h) \wedge |\lambda(h) - \lambda(h')| = r\})$ is contained in the set of points within distance $sin(\pi r/\lambda(h)) \leq \pi r/\lambda$ of the two hyperplanes representing $h_1(x) = sign(w \cdot x + b_1)$ and $h_2(x) = sign(w \cdot x + b_2)$ defined by the property that $\lambda(h_1) - \lambda(h) = \lambda(h) - \lambda(h_2) = r$, so that the total region of disagreement is contained within

$$\{x : h_1(x) \neq h_2(x)\} \cup \{x : \min\{|w \cdot x + b_1|, |w \cdot x + b_2|\} \leq \pi r/\lambda(h)\}.$$

Clearly, $\mathbb{P}(\{x : h_1(x) \neq h_2(x)\}) = 2r$. Using previous results [2, 15], we know that $\mathbb{P}(\{x : \min\{|w \cdot x + b_1|, |w \cdot x + b_2|\} \leq \pi r/\lambda(h)\}) \leq 2\pi\sqrt{n}r/\lambda(h)$ (since the probability mass contained within this distance of a hyperplane is maximized when the hyperplane passes through the origin). Thus, the probability of the entire region of disagreement is at most $(2 + 2\pi\sqrt{n}/\lambda(h))r$, so that (5.1) holds, and therefore the disagreement coefficient is finite. □

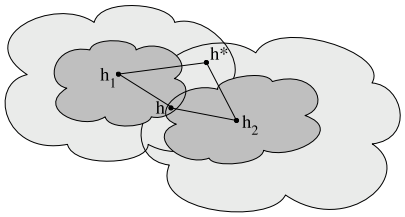


**Fig. 5.2.** Illustration of the proof of Theorem 5. The dark gray regions represent $B_{D_1}(h_1, 2r)$ and $B_{D_2}(h_2, 2r)$. The function $h$ that gets returned is in the intersection of these. The light gray regions represent $B_{D_1}(h_1, \epsilon/3)$ and $B_{D_2}(h_2, \epsilon/3)$. The target function $h^*$ is in the intersection of these. We therefore must have $r \leq \epsilon/3$, and by the triangle inequality $\mathrm{er}(h) \leq \epsilon$.

### 5.3 Composition results

We can also extend the results from the previous subsection to other types of distributions and concept classes in a variety of ways. Here we include a few results to this end.

**Close distributions:** If $(C, D)$ is learnable at an exponential rate, then for any distribution $D'$ such that for all measurable $A \subseteq \mathcal{X}$, $\lambda \mathbb{P}_D(A) \leq \mathbb{P}_{D'}(A) \leq (1/\lambda)\mathbb{P}_D(A)$ for some $\lambda \in (0, 1]$, $(C, D')$ is also learnable at an exponential rate. In particular, we can simply use the algorithm for $(C, D)$, filter the examples from $D'$ so that they appear like examples from $D$, and then any $t$ large enough to find an $\epsilon\lambda$-good classifier with respect to $D$ is large enough to find an $\epsilon$-good classifier with respect to $D'$. This general idea has previously been observed for the verifiable sample complexity [10, 15].

**Mixtures of distributions:** Suppose there exist algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ for learning a class $C$ at an exponential rate under distributions $D_1$ and $D_2$ respectively. It turns out we can also learn under any *mixture* of $D_1$ and $D_2$ at an exponential rate, by using $\mathcal{A}_1$ and $\mathcal{A}_2$ as black boxes. In particular, the following theorem relates the sample complexity under a mixture to the sample complexities under the mixing components.

**Theorem 5.** *Let $C$ be an arbitrary hypothesis class. Assume that the pairs $(C, D_1)$ and $(C, D_2)$ have sample complexities $S_1(\epsilon, \delta, h^*)$ and $S_2(\epsilon, \delta, h^*)$ respectively, where $D_1$ and $D_2$ have density functions $\mathbb{P}_{D_1}$ and $\mathbb{P}_{D_2}$ respectively. Then for any $\alpha \in [0, 1]$, the pair $(C, \alpha D_1 + (1 - \alpha)D_2)$ has sample complexity at most $2\lceil \max\{S_1(\epsilon/3, \delta/2, h^*), S_2(\epsilon/3, \delta/2, h^*)\}\rceil$.*

*Proof.* If $\alpha = 0$ or $1$ then the theorem statement holds trivially. Assume instead that $\alpha \in (0, 1)$. As we are only interested in proving the existence of an algorithm achieving the desired sample complexity, we can describe a method in terms of $\alpha$, $D_1$, and $D_2$, and in particular it can depend on these items in essentially arbitrary ways.

Suppose algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ achieve the stated sample complexities under $D_1$ and $D_2$ respectively. At a high level, the algorithm we define works by "filtering" the

distribution over input so that it appears to come from two streams, one distributed according to $D_1$, and one distributed according to $D_2$, and feeding these filtered streams to $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively. To do so, we define a random sequence $u_1, u_2, \cdots$ of independent uniform random variables in $[0, 1]$. We then run $\mathcal{A}_1$ on the sequence of examples $x_i$ from the unlabeled data sequence satisfying

$$u_i < \frac{\alpha \mathbb{P}_{D_1}(x_i)}{\alpha \mathbb{P}_{D_1}(x_i) + (1-\alpha)\mathbb{P}_{D_2}(x_i)},$$

and run $\mathcal{A}_2$ on the remaining examples, allowing each to make an equal number of label requests.

Let $h_1$ and $h_2$ be the classifiers output by $\mathcal{A}_1$ and $\mathcal{A}_2$. Because of the filtering, the examples that $\mathcal{A}_1$ sees are distributed according to $D_1$, so after $t/2$ queries, the current error of $h_1$ with respect to $D_1$ is, with probability $1 - \delta/2$, at most $\inf\{\epsilon' : S_1(\epsilon', \delta/2, h^*) \leq t/2\}$. A similar argument applies to the error of $h_2$ with respect to $D_2$.

Finally, let

$$r = \inf\{r : B_{D_1}(h_1, r) \cap B_{D_2}(h_2, r) \neq \emptyset\},$$

where

$$B_{D_i}(h_i, r) = \{h \in C : \mathbb{P}_{D_i}(h(x) \neq h_i(x)) \leq r\}.$$

Define the output of the algorithm to be any $h \in B_{D_1}(h_1, 2r) \cap B_{D_2}(h_2, 2r)$. If a total of $t \geq 2\lceil \max\{S_1(\epsilon/3, \delta/2, h^*), S_2(\epsilon/3, \delta/2, h^*)\}\rceil$ queries have been made ($t/2$ by $\mathcal{A}_1$ and $t/2$ by $\mathcal{A}_2$), then by a union bound, with probability at least $1 - \delta$, $h^*$ is in the intersection of the $\epsilon/3$-balls, and so $h$ is in the intersection of the $2\epsilon/3$-balls. By the triangle inequality, $h$ is within $\epsilon$ of $h^*$ under both distributions, and thus also under the mixture. (See Figure 5.2 for an illustration of these ideas.) □

### 5.4 Lower Bounds

Given the previous discussion, one might suspect that *any* pair $(C, D)$ is learnable at an exponential rate, under some mild condition such as finite VC dimension. However, we show in the following that this is *not* the case, even for some simple geometric concept classes when the distribution is especially nasty.

**Theorem 6.** *For any positive function $\phi(\epsilon) = o(1/\epsilon)$, there exists a pair $(C, D)$, with the VC dimension of $C$ equal 1, such that for any achievable sample complexity $S(\epsilon, \delta, h)$ for $(C, D)$, for any $\delta \in (0, 1/4)$,*

$$\exists h \in C \text{ s.t. } S(\epsilon, \delta, h) \neq o(\phi(\epsilon)).$$

*In particular, taking $\phi(\epsilon) = 1/\sqrt{\epsilon}$ (for example), this implies that there exists a $(C, D)$ that is not learnable at an exponential rate (in the sense of Definition 4).*

*Proof.* If we can prove this for any such $\phi(\epsilon) \neq O(1)$, then clearly this would imply the result holds for $\phi(\epsilon) = O(1)$ as well, so we will focus on $\phi(\epsilon) \neq O(1)$ case. Let $T$ be a fixed infinite tree in which each node at depth $i$ has $c_i$ children; $c_i$ is defined shortly

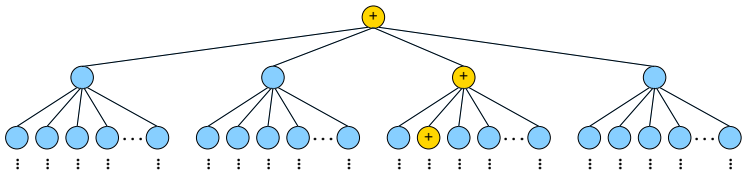

**Fig. 5.3.** A learning problem where exponential rates are not achievable. The instance space is an infinite-depth tree. The target labels nodes along a single infinite path as $+1$, and labels all other nodes $-1$. For any $\phi(\epsilon) = o(1/\epsilon)$, when the number of children and probability mass of each node at each subsequent level are set in a certain way, sample complexities of $o(\phi(\epsilon))$ are not achievable for all targets.

below. We consider learning the hypothesis class $C$ where each $h \in C$ corresponds to a path down the tree starting at the root; every node along this path is labeled $1$ while the remaining nodes are labeled $-1$. Clearly for each $h \in C$ there is precisely one node on each level of the tree labeled $1$ by $h$ (i.e. one node at each depth). $C$ has VC dimension $1$ since knowing the identity of the node labeled $1$ on level $i$ is enough to determine the labels of all nodes on levels $0, \ldots, i$ perfectly. This learning problem is depicted in Figure 5.3.

Now we define $D$, a "bad" distribution for $C$. Let $\{\ell_i\}_{i=1}^{\infty}$ be any sequence of positive numbers s.t. $\sum_{i=1}^{\infty} \ell_i = 1$. $\ell_i$ will bound the total probability of all nodes on level $i$ according to $D$. Assume all nodes on level $i$ have the same probability according to $D$, and call this $p_i$. We define the values of $p_i$ and $c_i$ recursively as follows. For each $i \geq 1$, we define $p_i$ as any positive number s.t. $p_i \lceil \phi(p_i) \rceil \prod_{j=0}^{i-2} c_j \leq \ell_i$ and $\phi(p_i) \geq 4$, and define $c_{i-1} = \lceil \phi(p_i) \rceil$. We are guaranteed that such a value of $p_i$ exists by the assumptions that $\phi(\epsilon) = o(1/\epsilon)$, meaning $\lim_{\epsilon \to 0} \epsilon \phi(\epsilon) = 0$, and that $\phi(\epsilon) \neq O(1)$. Letting $p_0 = 1 - \sum_{i \geq 1} p_i \prod_{j=0}^{i-1} c_j$ completes the definition of $D$.

With this definition of the parameters above, since $\sum_i p_i \leq 1$, we know that for any $\epsilon_0 > 0$, there exists some $\epsilon < \epsilon_0$ such that for some level $j$, $p_j = \epsilon$ and thus $c_{j-1} \geq \phi(p_j) = \phi(\epsilon)$. We will use this fact to show that $\propto \phi(\epsilon)$ labels are needed to learn with error less than $\epsilon$ for these values of $\epsilon$. To complete the proof, we must prove the existence of a "difficult" target function, customized to challenge the particular learning algorithm being used. To accomplish this, we will use the probabilistic method to prove the existence of a point in each level $i$ such that any target function labeling that point positive would have a sample complexity $\geq \phi(p_i)/4$. Furthermore, we will show that we can find such a point at each level in a recursive manner, so that the point at level $i$ is among the children of the point at level $i - 1$. Then the difficult target function simply strings these points together.

To begin, we define $x_0 =$ the root node. Then for each $i \geq 1$, recursively define $x_i$ as follows. Suppose, for any $h$, the set $R_h$ and the classifier $\hat{h}_h$ are, respectively, the random variable representing the set of examples the learning algorithm would request, and the classifier the learning algorithm would output, when $h$ is the target and its label request budget is set to $t = \lfloor \phi(p_i)/2 \rfloor$. For any node $x$, we will let Children$(x)$ denote the set of children of $x$, and Subtree$(x)$ denote the set of $x$ along with all descendants of $x$. Additionally, let $h_x$ denote any classifier in $C$ s.t. $h_x(x) = +1$. Also, for two

classifiers $h_1, h_2$, define $er(h_1; h_2) = \mathbb{P}_D(h_1(X) \neq h_2(X))$. Now note that

$$\begin{aligned}
&\max_{x \in \text{Children}(x_{i-1})} \inf_{h \in C: h(x)=+1} \mathbb{P}\{er(\hat{h}_h; h) > p_i\} \\
&\geq \frac{1}{c_{i-1}} \sum_{x \in \text{Children}(x_{i-1})} \inf_{h \in C: h(x)=+1} \mathbb{P}\{er(\hat{h}_h; h) > p_i\} \\
&\geq \frac{1}{c_{i-1}} \sum_{x \in \text{Children}(x_{i-1})} \mathbb{P}\{\forall h \in C : h(x) = +1, \text{Subtree}(x) \cap R_h = \emptyset \wedge er(\hat{h}_h; h) > p_i\} \\
&= \mathbb{E}\left[\frac{1}{c_{i-1}} \sum_{\substack{x \in \text{Children}(x_{i-1}): \\ \text{Subtree}(x) \cap R_{h_x} = \emptyset}} \mathbb{I}\left[\forall h \in C : h(x) = +1, er(\hat{h}_h; h) > p_i\right]\right] \\
&\geq \mathbb{E}\left[\min_{x' \in \text{Children}(x_{i-1})} \frac{1}{c_{i-1}} \sum_{\substack{x \in \text{Children}(x_{i-1}): \\ \text{Subtree}(x) \cap R_{h_x} = \emptyset}} \mathbb{I}\left[x' \neq x\right]\right] \\
&\geq \frac{1}{c_{i-1}}(c_{i-1} - t - 1) = \frac{1}{\lfloor \phi(p_i) \rfloor}(\lfloor \phi(p_i) \rfloor - \lfloor \phi(p_i)/2 \rfloor - 1) \\
&\geq \frac{1}{\lfloor \phi(p_i) \rfloor}(\lfloor \phi(p_i) \rfloor / 2 - 1) \geq 1/4.
\end{aligned}$$

The expectations above are over the unlabeled examples and any internal random bits used by the algorithm. The above inequalities imply there exists some $x \in \text{Children}(x_{i-1})$ such that every $h \in C$ that has $h(x) = +1$ has $S(p_i, \delta, h) \geq \lfloor \phi(p_i)/2 \rfloor \geq \phi(p_i)/4$; we will take $x_i$ to be this value of $x$. We now simply take the target function $h^*$ to be the classifier that labels $x_i$ positive for all $i$, and labels every other point negative. By construction, we have $\forall i, S(p_i, \delta, h^*) \geq \phi(p_i)/4$, and therefore

$$\forall \epsilon_0 > 0, \exists \epsilon < \epsilon_0 : S(\epsilon, \delta, h^*) \geq \phi(\epsilon)/4,$$

so that $S(\epsilon, \delta, h^*) \neq o(\phi(\epsilon))$. □

Note that this implies that the $o(1/\epsilon)$ guarantee of Corollary 1 is in some sense the tightest guarantee we can make at that level of generality, without using a more detailed description of the structure of the problem beyond the finite VC dimension assumption.

This type of example can be realized by certain nasty distributions, even for a variety of simple hypothesis classes: for example, linear separators in $\mathbb{R}^2$ or axis-aligned rectangles in $\mathbb{R}^2$. We remark that this example can also be modified to show that we cannot expect intersections of classifiers to preserve exponential rates. That is, the proof can be extended to show that there exist classes $C_1$ and $C_2$, such that both $(C_1, D)$ and $(C_2, D)$ are learnable at an exponential rate, but $(C, D)$ is not, where $C = \{h_1 \cap h_2 : h_1 \in C_1, h_2 \in C_2\}$.

## 6 Discussion and Open Questions

The implication of our analysis is that in many interesting cases where it was previously believed that active learning could not help, it turns out that active learning *does help asymptotically*. We have formalized this idea and illustrated it with a number of examples and general theorems throughout the paper. This realization dramatically shifts our understanding of the usefulness of active learning: while previously it was thought that active learning could *not* provably help in any but a few contrived and unrealistic learning problems, in this alternative perspective we now see that active learning essentially *always* helps, and does so significantly in all *but* a few contrived and unrealistic problems.

The use of decompositions of $C$ in our analysis generates another interpretation of these results. Specifically, Dasgupta [10] posed the question of whether it would be useful to develop active learning techniques for looking at unlabeled data and "placing bets" on certain hypotheses. One might interpret this work as an answer to this question; that is, some of the decompositions used in this paper can be interpreted as reflecting a preference partial-ordering of the hypotheses, similar to ideas explored in the passive learning literature [21, 19, 3]. However, the construction of a good decomposition in active learning seems more subtle and quite different from previous work in the context of supervised or semi-supervised learning.

It is interesting to examine the role of target- and distribution-dependent constants in this analysis. As defined, both the verifiable and true sample complexities may depend heavily on the particular target function and distribution. Thus, in both cases, we have interpreted these quantities as fixed when studying the asymptotic growth of these sample complexities as $\epsilon$ approaches $0$. It has been known for some time that, with only a few unusual exceptions, any target- and distribution-independent bound on the verifiable sample complexity could typically be no better than the sample complexity of passive learning; in particular, this observation lead Dasgupta to formulate his splitting index bounds as both target- and distribution-dependent [10]. This fact also applies to bounds on the true sample complexity as well. Indeed, the entire distinction between verifiable and true sample complexities collapses if we remove the dependence on these unobservable quantities.

One might wonder what the practical implications of the true sample complexity of active learning might be since the theoretical improvements we provide are for an unverifiable complexity measure and therefore they do not actually inform the user (or algorithm) of how many labels to allow the algorithm to request. However, there might still be implications for the design of practical algorithms. In some sense, this is the same issue faced in the analysis of universally consistent learning rules in passive learning [13]. There is typically no way to verify how close to the Bayes error rate a classifier is (verifiable complexity is infinite), yet we still want learning rules whose error rates provably converge to the Bayes error in the limit (true complexity is a finite function of epsilon and the distribution of $(X, Y)$), and we often find such methods quite effective in practice (e.g., $k$-nearest neighbor methods). So this is one instance where an unverifiable sample complexity seems to be a useful guide in algorithm design. In active learning with finite-complexity hypothesis classes we are more fortunate, since the verifiable complexity is finite – and we certainly want algorithms with small

verifiable sample complexity; however, an analysis of unverifiable complexities still seems relevant, particularly when the verifiable complexity is large. In general, it seems desirable to design algorithms for any given active learning problem that achieve both a verifiable sample complexity that is near optimal and a true sample complexity that is asymptotically better than passive learning.

**Open Questions:** There are many interesting open problems within this framework. Perhaps the most interesting of these would be formulating general necessary and sufficient conditions for learnability at an exponential rate, and determining whether Theorem 1 can be extended to the agnostic case or to infinite capacity hypothesis classes.

**Subsequent Work:** Since the initial publication of these results at the 2008 Conference on Learning Theory [5], there has been some progress in this area worth reporting. In particular, recall that Theorem 1 allows the algorithm to depend on the distribution $D$ in arbitrary ways; that is, for each $D$, we can use a different active learning algorithm to achieve the improvements over the passive learning method. Indeed the proof of this result presented in Appendix E employs an active learning algorithm that leverages this dependence to such an extent that it does not seem feasible to remove this dependence on the distribution without altering the fundamental nature of the algorithm. However, using an entirely different type of active learning algorithm, Hanneke [17] has recently been able to strengthen Theorem 1, proving that for any passive learning algorithm, there is an active learning algorithm achieving asymptotically strictly superior sample complexities simultaneously for all nontrivial target functions *and* distributions.

**Acknowledgments:** We would like to extend a special thanks to Yishay Mansour for introducing us to this perspective on sample complexity and for helping us to initiate this line of thought. We are also grateful to Eyal Even-Dar, Michael Kearns, Larry Wasserman, and Eric Xing for their helpful feedback.
Maria-Florina has been supported in part by an IBM Graduate Fellowship, a Google Research Grant, and by ONR grant N00014-09-1-0751. Steve has been supported by an IBM Graduate Fellowship and by the NSF grant IIS-0713379 awarded to Eric Xing.

## References

1. A. Antos and G. Lugosi. Strong minimax lower bounds for learning. *Machine Learning*, 30:31–56, 1998.
2. M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.
3. M.-F. Balcan and A. Blum. A PAC-style model for learning from labeled and unlabeled data. Book chapter in "Semi-Supervised Learning", O. Chapelle and B. Schlkopf and A. Zien, eds., MIT press, 2006.
4. M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Proceedings of the 20th Annual Conference on Learning Theory*, 2007.
5. M.-F. Balcan, S. Hanneke, and J. Wortman. The true sample complexity of active learning. In *Proceedings of the 21st Annual Conference on Learning Theory*, 2008.
6. A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

7. R. Castro and R. Nowak. Minimax bounds for active learning. In *Proceedings of the 20th Annual Conference on Learning Theory*, 2007.
8. D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.
9. S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems*, 2004.
10. S. Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems*, 2005.
11. S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems*, 2007.
12. S. Dasgupta, A. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.
13. L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.
14. Y. Freund, H.S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.
15. S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.
16. S. Hanneke. Teaching dimension and the complexity of active learning. In *Proceedings of the 20th Conference on Learning Theory*, 2007.
17. S. Hanneke. *Theoretical Foundations of Active Learning*. PhD thesis, Machine Learning Department, Carnegie Mellon University, Forthcoming.
18. D. Haussler, N. Littlestone, and M. Warmuth. Predicting $\{0,1\}$-functions on randomly drawn points. *Information and Computation*, 115:248–292, 1994.
19. J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.
20. V. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York, 1982.
21. V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons Inc., 1998.

# Appendix

## A A Lower Bound on the Verifiable Sample Complexity of the Empty Interval

Let $h_-$ denote the all-negative interval. In this section, we lower bound the verifiable sample complexity achievable for this classifier, with respect to the hypothesis class $C$ of interval classifiers under a uniform distribution on $[0,1]$. Specifically, suppose there exists an algorithm $A$ that achieves a verifiable sample complexity $S(\epsilon, \delta, h)$ such that for some $\epsilon \in (0, 1/4)$ and some $\delta \in (0, 1/4)$,

$$S(\epsilon, \delta, h_-) < \left\lfloor \frac{1}{24\epsilon} \right\rfloor .$$

We prove that this would imply the existence of some interval $h'$ for which the value of $S(\epsilon, \delta, h')$ is *not valid* under Definition 1. We proceed by the probabilistic method.

Consider the subset of intervals

$$H_\epsilon = \left\{ [3i\epsilon, 3(i+1)\epsilon] : i \in \left\{ 0, 1, \ldots, \left\lfloor \frac{1-3\epsilon}{3\epsilon} \right\rfloor \right\} \right\} .$$

Let $s = \lceil S(\epsilon, \delta, h_-) \rceil$. For any $f \in C$, let $R_f$, $\hat{h}_f$, and $\hat{\epsilon}_f$ denote the random variables representing, respectively, the set of examples $(x, y)$ for which $A(s, \delta)$ requests labels (including their $y = f(x)$ labels), the classifier $A(s, \delta)$ outputs, and the confidence bound $A(s, \delta)$ outputs, when $f$ is the target function. Let $\mathbb{I}$ be an indicator function that is 1 if its argument is true and 0 otherwise. Then

$$\begin{aligned}
&\max_{f \in H_\epsilon} \mathbb{P}\left( \mathbb{P}_X \left( \hat{h}_f(X) \neq f(X) \right) > \hat{\epsilon}_f \right) \\
&\quad \geq \frac{1}{|H_\epsilon|} \sum_{f \in H_\epsilon} \mathbb{P}\left( \mathbb{P}_X \left( \hat{h}_f(X) \neq f(X) \right) > \hat{\epsilon}_f \right) \\
&\quad \geq \frac{1}{|H_\epsilon|} \sum_{f \in H_\epsilon} \mathbb{P}\left( (R_f = R_{h_-}) \wedge \left( \mathbb{P}_X \left( \hat{h}_f(X) \neq f(X) \right) > \hat{\epsilon}_f \right) \right) \\
&\quad = \mathbb{E}\left[ \frac{1}{|H_\epsilon|} \sum_{f \in H_\epsilon : R_f = R_{h_-}} \mathbb{I}\left[ \mathbb{P}_X \left( \hat{h}_f(X) \neq f(X) \right) > \hat{\epsilon}_f \right] \right] \\
&\quad \geq \mathbb{E}\left[ \frac{1}{|H_\epsilon|} \sum_{f \in H_\epsilon : R_f = R_{h_-}} \mathbb{I}\left[ \left( \mathbb{P}_X \left( \hat{h}_f(X) = +1 \right) \leq \epsilon \right) \wedge (\hat{\epsilon}_f \leq \epsilon) \right] \right] \qquad \text{(A.1)} \\
&\quad = \mathbb{E}\left[ \frac{1}{|H_\epsilon|} \sum_{f \in H_\epsilon : R_f = R_{h_-}} \mathbb{I}\left[ \left( \mathbb{P}_X \left( \hat{h}_{h_-}(X) \neq h_-(X) \right) \leq \epsilon \right) \wedge \left( \hat{\epsilon}_{h_-} \leq \epsilon \right) \right] \right] \qquad \text{(A.2)} \\
&\quad \geq \mathbb{E}\left[ \left( \frac{|H_\epsilon| - s}{|H_\epsilon|} \right) \mathbb{I}\left[ \mathbb{P}_X \left( \hat{h}_{h_-}(X) \neq h_-(X) \right) \leq \hat{\epsilon}_{h_-} \leq \epsilon \right] \right] \qquad \text{(A.3)} \\
&\quad = \left( \frac{|H_\epsilon| - s}{|H_\epsilon|} \right) \mathbb{P}\left( \mathbb{P}_X \left( \hat{h}_{h_-}(X) \neq h_-(X) \right) \leq \hat{\epsilon}_{h_-} \leq \epsilon \right) \\
&\quad \geq \left( \frac{|H_\epsilon| - s}{|H_\epsilon|} \right) (1 - \delta) > \delta.
\end{aligned}$$

All expectations are over the draw of the unlabeled examples and any additional random bits used by the algorithm. Line A.1 follows from the fact that all intervals $f \in H_\epsilon$ are of width $3\epsilon$, so if $\hat{h}_f$ labels less than a fraction $\epsilon$ of the points as positive, it must make an error of at least $2\epsilon$ with respect to $f$, which is more than $\hat{\epsilon}_f$ if $\hat{\epsilon}_f \leq \epsilon$. Note that, for any fixed sequence of unlabeled examples and additional random bits used by the algorithm, the sets $R_f$ are completely determined, and any $f$ and $f'$ for which $R_f = R_{f'}$ must have $\hat{h}_f = \hat{h}_{f'}$ and $\hat{\epsilon}_f = \hat{\epsilon}_{f'}$. In particular, any $f$ for which $R_f = R_{h_-}$ will yield identical outputs from the algorithm, which implies line A.2. Furthermore, the only classifiers $f \in H_\epsilon$ for which $R_f \neq R_{h_-}$ are those for which some $(x, -1) \in R_{h_-}$ has $f(x) = +1$ (i.e., $x$ is in the $f$ interval). But since there is zero probability that any

unlabeled example is in more than one of the intervals in $H_\epsilon$, with probability 1 there are at most $s$ intervals $f \in H_\epsilon$ with $R_f \neq R_{h^-}$, which explains line A.3.

This proves the existence of some target function $h^* \in C$ such that $\mathbb{P}(er(h_{s,\delta}) > \hat{\epsilon}_{s,\delta}) > \delta$, which contradicts the conditions of Definition 1. □

## B Proof of Theorem 2

First note that the total number of label requests used by the aggregation procedure in Algorithm 1 is at most $t$. Initially running the algorithms $A_1, \ldots, A_k$ requires $\sum_{i=1}^{k} \lfloor t/(4i^2) \rfloor \leq t/2$ labels, and the second phase of the algorithm requires $k^2 \lceil 72 \ln(4k/\delta) \rceil$ labels, which by definition of $k$ is also less than $t/2$. Thus this procedure is a valid learning algorithm.

Now suppose that the true target $h^*$ is a member of $C_i$. We must show that for any input $t$ such that

$$t \geq \max \left\{ 4i^2 \left\lceil S_i(\epsilon/2, \delta/2, h^*) \right\rceil, 2i^2 \left\lceil 72 \ln(4i/\delta) \right\rceil \right\} ,$$

the aggregation procedure outputs a hypothesis $\hat{h}_t$ such that $er(\hat{h}_t) \leq \epsilon$ with probability at least $1 - \delta$.

First notice that since $t \geq 2i^2 \lceil 72 \ln(4i/\delta) \rceil$, $k \geq i$. Furthermore, since $t/(4i^2) \geq \lceil S_i(\epsilon/2, \delta/2, h^*) \rceil$, with probability at least $1 - \delta/2$, running $\mathcal{A}_i(\lfloor t/(4i^2) \rfloor, \delta/2)$ returns a function $h_i$ with $er(h_i) \leq \epsilon/2$.

Let $j^* = \operatorname{argmin}_j er(h_j)$. Since $er(h_{j*}) \leq er(h_\ell)$ for any $\ell$, we would expect $h_{j^*}$ to make no more errors that $h_\ell$ on points where the two functions disagree. It then follows from Hoeffding's inequality, with probability at least $1 - \delta/4$, for all $\ell$,

$$m_{j^*\ell} \leq \frac{7}{12} \lceil 72 \ln (4k/\delta) \rceil ,$$

and thus

$$\min_j \max_\ell m_{j\ell} \leq \frac{7}{12} \lceil 72 \ln(4k/\delta) \rceil .$$

Similarly, by Hoeffding's inequality and a union bound, with probability at least $1 - \delta/4$, for any $\ell$ such that

$$m_{\ell j^*} \leq \frac{7}{12} \lceil 72 \ln(4k/\delta) \rceil ,$$

the probability that $h_\ell$ mislabels a point $x$ given that $h_\ell(x) \neq h_{j^*}(x)$ is less than $2/3$, and thus $er(h_\ell) \leq 2er(h_{j^*})$. By a union bound over these three events, we find that, as desired, with probability at least $1 - \delta$,

$$er(\hat{h}_t) \leq 2er(h_{j^*}) \leq 2er(h_i) \leq \epsilon .$$

□

## C Proof of Theorem 3

Assume that $(C, D)$ is learnable at an exponential rate. This means that there exists an algorithm $A$ such that for any target $h^*$ in $C$, there exist constants $\gamma_{h^*}$ and $k_{h^*}$ such that for any $\epsilon$ and $\delta$, with probability at least $1-\delta$, for any $t \geq \gamma_{h^*}(\log(1/(\epsilon\delta)))^{k_{h^*}}$, after $t$ label requests, $A(t, \delta)$ outputs an $\epsilon$-good classifier.

For each $i$, let

$$C_i = \{h \in C : \gamma_h \leq i, k_h \leq i\} .$$

Define an algorithm $A_i$ that achieves the required polylog verifiable sample complexity on $(C_i, D)$ as follows. First, run the algorithm $A$ to obtain a function $h_A$. Then, output the classifier in $C_i$ that is *closest to* $h_A$, i.e., the classifier that minimizes the probability of disagreement with $h_A$. If $t \geq i(\log(2/(\epsilon\delta)))^i$, then after $t$ label requests, with probability at least $1-\delta$, $A(t, \delta)$ outputs an $\epsilon/2$-good classifier, so by the triangle inequality, with probability at least $1-\delta$, $A_i(t, \delta)$ outputs an $\epsilon$-good classifier.

It can be guaranteed that with probability at least $1-\delta$, the function output by $A_i$ has error no more than $\hat{\epsilon}_t = (2/\delta)\exp\left\{-(t/i)^{1/i}\right\}$, which is no more than $\epsilon$, implying that the expression above is a *verifiable* sample complexity.

Combining this with Theorem 2 yields the desired result. □

## D Heuristic Approaches to Decomposition

As mentioned, decomposing purely based on verifiable complexity with respect to $(C, D)$ typically cannot yield a good decomposition even for very simple problems, such as unions of intervals (see Section 5.2). The reason is that the set of classifiers with high verifiable sample complexity may itself have high verifiable complexity.

Although we have not yet found a general method that can provably always find a good decomposition when one exists (other than the trivial method in the proof of Theorem 3), we find that a heuristic recursive technique is frequently effective. To begin, define $C_1 = C$. Then for $i > 1$, recursively define $C_i$ as the set of all $h \in C_{i-1}$ such that $\theta_h = \infty$ with respect to $(C_{i-1}, D)$. (Here $\theta_h$ is the disagreement coefficient of $h$; see Definition 3.) Suppose that for some $N$, $C_{N+1} = \emptyset$. Then for the decomposition $C_1, C_2, \ldots, C_N$, every $h \in C$ has $\theta_h < \infty$ with respect to at least one of the sets in which it is contained, which implies that the verifiable sample complexity of $h$ with respect to that set is $O(\mathrm{polylog}(1/\epsilon\delta))$, and the aggregation algorithm can be used to achieve polylog sample complexity.

We could alternatively perform a similar decomposition using a suitable definition of splitting index [10], or more generally using

$$\limsup_{\epsilon \to 0} \frac{S_{C_{i-1}}(\epsilon, \delta, h)}{\left(\log\left(\frac{1}{\epsilon\delta}\right)\right)^k}$$

for some fixed constant $k > 0$.

This procedure does not always generate a good decomposition. However, if $N < \infty$ exists, then it creates a decomposition for which the aggregation algorithm, combined with an appropriate sequence of algorithms $\{A_i\}$, could achieve exponential

rates. In particular, this is the case for all of the $(C, D)$ described in Section 5. In fact, even if $N = \infty$, as long as every $h \in C$ does end up in *some* set $C_i$ for finite $i$, this decomposition would still provide exponential rates.

## E Proof of Theorem 1

We now finally prove Theorem 1. This section is mostly self-contained, though we do make use of Theorem 2 from Section 4 in the final step of the proof.

The proof proceeds according to the following outline. We begin in Lemma 1 by describing special conditions under which a CAL-like algorithm has the property that the more unlabeled examples it considers, the smaller the fraction of them it asks to be labeled. Since CAL is able to identify the target's true label on any example it considers (either the label of the example is requested or the example is not in the region of disagreement and therefore the label is already known), we end up with a set of labeled examples growing strictly faster than the number of label requests used to obtain it. This set of labeled examples can be used as a training set in any passive learning algorithm. However, the special conditions under which this happens are rather limiting. In Lemma 2, we exploit a subtle relation between overlapping boundary regions and shatterable sets to show that we can decompose any finite VC dimension class into a countable number of subsets satisfying these special conditions. This, combined with the aggregation algorithm, and a simple procedure that boosts the confidence level, extends Lemma 1 to the general conditions of Theorem 1.

Before jumping into Lemma 1, it is useful to define some additional notation. For any $V \subseteq C$ and $h \in C$, define

$$\tilde{B}_V(h, r) = \left\{h' \in \tilde{V} : \mathbb{P}_D(h(x) \neq h'(x)) \leq r\right\} ,$$

where $\tilde{V}$ is a countable dense subset of $V$.[7] Define the *boundary* of $h$ with respect to $D$ and $V$, denoted $\partial_V h$, as

$$\partial_V h = \lim_{r \to 0} \mathrm{DIS}(\tilde{B}_V(h, r)).$$

**Lemma 1.** *Suppose $(C, D)$ is such that $C$ has finite VC dimension $d$, and $\forall h \in C, \mathbb{P}(\partial_C h) = 0$. Then for any passive learning sample complexity $S_p(\epsilon, \delta, h)$ for $(C, D)$ which is nondecreasing as $\epsilon \to 0$, there exists an active learning algorithm achieving a sample complexity $S_a(\epsilon, \delta, h)$ such that, for any $\delta > 0$ and any target function $h^* \in C$ with $S_p(\epsilon, \delta, h^*) = \omega(1)$ and $\forall \epsilon > 0$,$S_p(\epsilon, \delta, h^*) < \infty$,*

$$S_a(\epsilon, 2\delta, h^*) = o(S_p(\epsilon, \delta, h^*)) .$$

*Proof.* Recall that $t$ is the "budget" of the active learning algorithm, and our goal in this proof is to define an active learning algorithm $A_a$ and a function $S_a(\epsilon, \delta, h^*)$ such that, if $t \geq S_a(\epsilon, \delta, h^*)$ and $h^* \in C$ is the target function, then $A_a(t, \delta)$ will, with probability $1 - \delta$, output an $\epsilon$-good classifier; furthermore, we require that $S_a(\epsilon, 2\delta, h^*) = o(S_p(\epsilon, \delta, h^*))$ under the conditions on $h^*$ in the lemma statement.

[7] See the note in Definition 3.

To construct this algorithm, we perform the learning in two phases. The first is a passive phase, where we focus on reducing a version space, to shrink the region of disagreement; the second is a phase where we construct a labeled training set, which is much larger than the number of label requests used to construct it since all classifiers in the version space agree on many of the examples' labels.

To begin the first phase, we simply request the labels of $x_1, x_2, \ldots, x_{\lfloor t/2 \rfloor}$, and let

$$V = \{h \in \tilde{C} : \forall i \leq \lfloor t/2 \rfloor, h(x_i) = h^*(x_i)\} .$$

In other words, $V$ is the set of all hypotheses in $\tilde{C}$ that correctly label the first $\lfloor t/2 \rfloor$ examples. By standard consistency results [20, 6, 13], there is a universal constant $c > 0$ such that, with probability at least $1 - \delta/2$,

$$\sup_{h \in V} er(h) \leq c \left( \frac{d \ln t + \ln \frac{1}{\delta}}{t} \right) .$$

This implies that

$$V \subseteq \tilde{B} \left( h^*, c \left( \frac{d \ln t + \ln \frac{1}{\delta}}{t} \right) \right) ,$$

and thus $\mathbb{P}(\mathrm{DIS}(V)) \leq \Delta_t$ where

$$\Delta_t = \mathbb{P} \left( \mathrm{DIS} \left( \tilde{B} \left( h^*, c \left( \frac{d \ln t + \ln \frac{1}{\delta}}{t} \right) \right) \right) \right) .$$

Clearly, $\Delta_t$ goes to 0 as $t$ grows, by the assumption on $\mathbb{P}(\partial_C h^*)$.

Next, in the second phase of the algorithm, we will actively construct a set of labeled examples to use with the passive learning algorithm. If ever we have $\mathbb{P}(\mathrm{DIS}(V)) = 0$ for some finite $t$, then clearly we can return any $h \in V$, so this case is easy.

Otherwise, let $n_t = \lfloor t/(24\mathbb{P}(\mathrm{DIS}(V)) \ln(4/\delta)) \rfloor$, and suppose $t \geq 2$. By a Chernoff bound, with probability at least $1 - \delta/2$, in the sequence of examples $x_{\lfloor t/2 \rfloor + 1}, x_{\lfloor t/2 \rfloor + 2}, \ldots, x_{\lfloor t/2 \rfloor + n_t}$, at most $t/2$ of the examples are in $\mathrm{DIS}(V)$. If this is not the case, we fail and output an arbitrary $h$; otherwise, we request the labels of every one of these $n_t$ examples that are in $\mathrm{DIS}(V)$.

Now construct a sequence $\mathcal{L} = \{(x'_1, y'_1), (x'_2, y'_2), \ldots, (x'_{n_t}, y'_{n_t})\}$ of labeled examples such that $x'_i = x_{\lfloor t/2 \rfloor + i}$, and $y'_i$ is either the label agreed upon by all the elements of $V$, or it is the $h^*(x_{\lfloor t/2 \rfloor + i})$ label value we explicitly requested. Note that because $\inf_{h \in V} er(h) = 0$ with probability 1, we also have that with probability 1 every $y'_i = h^*(x'_i)$. We may therefore use these $n_t$ examples as iid training examples for the passive learning algorithm.

Suppose $A$ is the passive learning algorithm that guarantees $S_p(\epsilon, \delta, h)$ passive sample complexities. Then let $h_t$ be the classifier returned by $A(\mathcal{L}, \delta)$. This is the classifier the active learning algorithm outputs.

Note that if $n_t \geq S_p(\epsilon, \delta, h^*)$, then with probability at least $1 - \delta$ over the draw of $\mathcal{L}$, $er(h_t) \leq \epsilon$. Define

$$S_a(\epsilon, 2\delta, h^*) = 1 + \inf \{s : s \geq 144 \ln(4/\delta) S_p(\epsilon, \delta, h^*) \Delta_s\} .$$

This is well-defined when $S_p(\epsilon, \delta, h^*) < \infty$ because $\Delta_s$ is nonincreasing in $s$, so some value of $s$ will satisfy the inequality. Note that if $t \geq S_a(\epsilon, 2\delta, h^*)$, then (with probability at least $1 - \delta/2$)

$$S_p(\epsilon, \delta, h^*) \leq \frac{t}{144 \ln(4/\delta) \Delta_t} \leq n_t \; .$$

So, by a union bound over the possible failure events listed above ($\delta/2$ for $\mathbb{P}(\mathrm{DIS}(V)) > \Delta_t$, $\delta/2$ for more than $t/2$ examples of $\mathcal{L}$ in $\mathrm{DIS}(V)$, and $\delta$ for $er(h_t) > \epsilon$ when the previous failures do not occur), if $t \geq S_a(\epsilon, 2\delta, h^*)$, then with probability at least $1 - 2\delta$, $er(h_t) \leq \epsilon$. So $S_a(\epsilon, \delta, h^*)$ is a valid sample complexity function, achieved by the described algorithm. Furthermore,

$$S_a(\epsilon, 2\delta, h^*) \leq 1 + 144 \ln(4/\delta) S_p(\epsilon, \delta, h^*) \Delta_{S_a(\epsilon, 2\delta, h^*)-2}.$$

If $S_a(\epsilon, 2\delta, h^*) = O(1)$, then since $S_p(\epsilon, \delta, h^*) = \omega(1)$, the result is established. Otherwise, since $S_a(\epsilon, \delta, h^*)$ is nondecreasing as $\epsilon \to 0$, $S_a(\epsilon, 2\delta, h^*) = \omega(1)$, so we know that $\Delta_{S_a(\epsilon, 2\delta, h^*)-2} = o(1)$. Thus, $S_a(\epsilon, 2\delta, h^*) = o\left(S_p(\epsilon, \delta, h^*)\right)$. □

As an interesting aside, it is also true (by essentially the same argument) that under the conditions of Lemma 1, the *verifiable* sample complexity of active learning is strictly smaller than the *verifiable* sample complexity of passive learning in this same sense. In particular, this implies a verifiable sample complexity that is $o\left(1/\epsilon\right)$ under these conditions. For instance, with some effort one can show that these conditions are satisfied when the VC dimension of $C$ is 1, or when the support of $D$ is at most countably infinite. However, for more complex learning problems, this condition will typically not be satisfied, and as such we require some additional work in order to use this lemma toward a proof of the general result in Theorem 1. Toward this end, we again turn to the idea of a decomposition of $C$, this time decomposing it into subsets satisfying the condition in Lemma 1.

In order to prove the existence of such a decomposition, we will rely on the assumption of finite VC dimension. The essential insight here is that anytime boundary regions are overlapping, we can shatter points in the overlap regions. To build the intuition for this, it may be helpful to first go through a related proof of a simpler (significantly weaker) result: namely, that not all classifiers in a $C$ with VC dimension 1 can have $\mathbb{P}(\partial_C h) = 1$.[8] This situation corresponds to all boundaries overlapping almost completely. To see that this is true, suppose the opposite, and consider any two classifiers $h_1, h_1' \in \tilde{C}$ with $\mathbb{P}(h_1(x) \neq h_1'(x)) > 0$. Let $A_1 = \{x : h_1(x) \neq h_1'(x)\}$ and $\Delta_1 = \mathbb{P}(A_1)/3$. Then, since $\mathbb{P}(\partial_C h_1 \cap \partial_C h_1') = 1$, there must be some region $A_1' \subseteq \partial_C h_1 \cap \partial_C h_1'$ with $\mathbb{P}(A_1') > 0$ and some $h_2 \in \tilde{B}_C(h_1, \Delta_1)$ for which $\mathbb{P}(x \in A_1' \wedge h_2(x) \neq h_1(x)) > 0$ (because $A_1' \subseteq \partial_C h_1$ and $\tilde{C}$ is countable). Furthermore, letting $A_1'' = \{x \in A_1' : h_2(x) \neq h_1(x)\}$, since $A_1'' \subseteq \partial_C h_1'$ there exists some $h_2' \in \tilde{B}_C(h_1', \Delta_1)$ with $\mathbb{P}(x \in A_1'' \wedge h_1'(x) \neq h_2'(x)) > 0$. Let $A_2 = \{x \in A_1'' : h_1'(x) \neq h_2'(x)\}$. Since, by construction, $\mathbb{P}(h_1(x) \neq h_2(x)) \leq \Delta_1$

[8] In fact, as mentioned, with a bit more effort, one can show that when the VC dimension is 1, *every* $h \in C$ has $\mathbb{P}(\partial_C h) = 0$. However, the weaker result studied here will be illustrative of a general technique applied in Lemma 2.

and $\mathbb{P}(h_1'(x) \neq h_2'(x)) \leq \Delta_1$, we have that $\{x : h_2(x) = h_1(x) \neq h_1'(x) = h_2'(x)\}$ is nonempty, so choose any point $x_1$ from this region. Furthermore, as mentioned, the set $A_2$ is nonempty, so take any point $x_2 \in A_2$. Then we have that $\{h_1, h_1', h_2, h_2'\}$ shatters $\{x_1, x_2\}$, which is a contradiction to the assumption that the VC dimension of $C$ is 1. It is not difficult to see that the argument can be applied repeatedly to add more points to the shatterable set, each time doubling the number of classifiers by finding another classifier sufficiently close to a respective classifier from the previous round so that it agrees with that one on most of all of the sets $A_1$, $A_2$, etc., but disagrees with that classifier on some subset of the overlap region of the boundaries of classifiers from the previous rounds.

This general relationship between overlapping boundaries and shatterable sets is the primary tool in proving the existence of a good decomposition. To extend the idea beyond the simple case of boundaries with probability one, we need some way to show that certain smaller boundaries will overlap under some conditions. To this end, we will prove that any set of classifiers that are sufficiently close together must have significant overlap in their boundaries, and thus if the boundaries have similar probabilities, the regions will be almost the same, and we can apply the above argument. The formal details are given below.

**Lemma 2.** *For any $(C, D)$ where $C$ has finite VC dimension $d$, there exists a countably infinite sequence $C_1, C_2, \ldots$ such that $C = \cup_{i=1}^{\infty} C_i$ and $\forall i, \forall h \in C_i, \mathbb{P}(\partial_{C_i} h) = 0$.*

*Proof.* The case of $d = 0$ is clear, so assume $d > 0$. A decomposition procedure is given in Algorithm 2. We will show that, if we let $\mathbb{H} = \text{Decompose}(C)$, then the maximum recursion depth is at most $d$ (counting the initial call as depth 0). Note that if this is true, then the lemma is proved, since it implies that $\mathbb{H}$ can be uniquely indexed by a $d$-tuple of integers, of which there are at most countably many.

**Algorithm 2** Decompose$(\mathcal{H})$

Let $\mathcal{H}_\infty = \{h \in \mathcal{H} : \mathbb{P}(\partial_{\mathcal{H}} h) = 0\}$
**if** $\mathcal{H}_\infty = \mathcal{H}$ **then**
  Return $\{\mathcal{H}\}$
**else**
  For $i \in \{1, 2, \ldots\}$, let $\mathcal{H}_i = \left\{h \in \mathcal{H} : \mathbb{P}(\partial_{\mathcal{H}} h) \in ((1 + 2^{-(d+3)})^{-i}, (1 + 2^{-(d+3)})^{1-i}]\right\}$
  Return $\bigcup_{i \in \{1,2,\ldots\}}$ Decompose$(\mathcal{H}_i) \cup \{\mathcal{H}_\infty\}$
**end if**

For the sake of contradiction, suppose that the maximum recursion depth of Decompose$(C)$ is more than $d$ (or is infinite). Thus, based on the first $d + 1$ recursive calls in one of those deepest paths in the recursion tree, there is a sequence of sets

$$C = \mathcal{H}^{(0)} \supseteq \mathcal{H}^{(1)} \supseteq \mathcal{H}^{(2)} \supseteq \cdots \mathcal{H}^{(d+1)} \neq \emptyset$$

and a corresponding sequence of finite positive integers $i_1, i_2, \ldots, i_{d+1}$ such that for each $j \in \{1, 2, \ldots, d+1\}$, every $h \in \mathcal{H}^{(j)}$ has

$$\mathbb{P}(\partial_{\mathcal{H}^{(j-1)}} h) \in \left( (1+2^{-(d+3)})^{-i_j}, (1+2^{-(d+3)})^{1-i_j} \right].$$

Take any $h_{d+1} \in \mathcal{H}^{(d+1)}$. There must exist some $r > 0$ such that $\forall j \in \{1, 2, \ldots, d+1\}$,

$$\mathbb{P}(\mathrm{DIS}(\tilde{B}_{\mathcal{H}^{(j-1)}}(h_{d+1}, r))) \in \left( (1+2^{-(d+3)})^{-i_j}, (1+2^{-(d+2)})(1+2^{-(d+3)})^{-i_j} \right]. \quad \text{(E.1)}$$

In particular, by (E.1), each $h \in \tilde{B}_{\mathcal{H}^{(j)}}(h_{d+1}, r/2)$ has

$$\mathbb{P}(\partial_{\mathcal{H}^{(j-1)}} h) > (1+2^{-(d+3)})^{-i_j} \geq (1+2^{-(d+2)})^{-1} \mathbb{P}(DIS(\tilde{B}_{\mathcal{H}^{(j-1)}}(h_{d+1}, r))),$$

though by definition of $\partial_{\mathcal{H}^{(j-1)}} h$ and the triangle inequality,

$$\mathbb{P}(\partial_{\mathcal{H}^{(j-1)}} h \setminus DIS(\tilde{B}_{\mathcal{H}^{(j-1)}}(h_{d+1}, r))) = 0.$$

Recall that in general, for sets $Q$ and $R_1, R_2, \ldots, R_k$, if $\mathbb{P}(R_i \setminus Q) = 0$ for all $i$, then $\mathbb{P}(\bigcap_i R_i) \geq \mathbb{P}(Q) - \sum_{i=1}^{k} (\mathbb{P}(Q) - \mathbb{P}(R_i))$. Thus, for any $j$, any set of $\leq 2^{d+1}$ classifiers $T \subset \tilde{B}_{\mathcal{H}^{(j)}}(h_{d+1}, r/2)$ must have

$$\mathbb{P}(\cap_{h \in T} \partial_{\mathcal{H}^{(j-1)}} h) \geq (1 - 2^{d+1}(1 - (1+2^{-(d+2)})^{-1})) \mathbb{P}(DIS(\tilde{B}_{\mathcal{H}^{(j-1)}}(h_{d+1}, r))) > 0.$$

That is, any set of $2^{d+1}$ classifiers in $\tilde{\mathcal{H}}^{(j)}$ within distance $r/2$ of $h_{d+1}$ will have boundaries with respect to $\mathcal{H}^{(j-1)}$ which have a nonzero probability overlap. The remainder of the proof will hinge on this fact that these boundaries overlap.

We now construct a shattered set of points of size $d+1$. Consider constructing a binary tree with $2^{d+1}$ leaves as follows. The root node contains $h_{d+1}$ (call this level $d+1$). Let $h_d \in \tilde{B}_{\mathcal{H}^{(d)}}(h_{d+1}, r/4)$ be some classifier with $\mathbb{P}(h_d(X) \neq h_{d+1}(X)) > 0$. Let the left child of the root be $h_{d+1}$ and the right child be $h_d$ (call this level $d$). Define $A_d = \{x : h_d(x) \neq h_{d+1}(x)\}$, and let $\Delta_d = 2^{-(d+2)} \mathbb{P}(A_d)$. Now for each $\ell \in \{d-1, d-2, \ldots, 0\}$ in decreasing order, we define the $\ell$ level of the tree as follows. Let $T_{\ell+1}$ denote the nodes at the $\ell+1$ level in the tree, and let $A'_\ell = \bigcap_{h \in T_{\ell+1}} \partial_{\mathcal{H}^{(\ell)}} h$. We iterate over the elements of $T_{\ell+1}$ in left-to-right order, and for each one $h$, we find $h' \in \tilde{B}_{\mathcal{H}^{(\ell)}}(h, \Delta_{\ell+1})$ with

$$\mathbb{P}_D(h(x) \neq h'(x) \wedge x \in A'_\ell) > 0 \,.$$

We then define the left child of $h$ to be $h$ and the right child to be $h'$, and we update

$$A'_\ell \leftarrow A'_\ell \cap \{x : h(x) \neq h'(x)\} \,.$$

After iterating through all the elements of $T_{\ell+1}$ in this manner, define $A_\ell$ to be the final value of $A'_\ell$ and $\Delta_\ell = 2^{-(d+2)} \mathbb{P}(A_\ell)$. The key is that, because every $h$ in the tree is within $r/2$ of $h_{d+1}$, the set $A'_\ell$ always has nonzero measure, and is contained in $\partial_{\mathcal{H}^{(\ell)}} h$ for any $h \in T_{\ell+1}$, so there always exists an $h'$ arbitrarily close to $h$ with $\mathbb{P}_D(h(x) \neq h'(x) \wedge x \in A'_\ell) > 0$.

Note that for $\ell \in \{0, 1, 2, \ldots, d\}$, every node in the left subtree of any $h$ at level $\ell + 1$ is strictly within distance $2\Delta_\ell$ of $h$, and every node in the right subtree of any $h$ at level $\ell + 1$ is strictly within distance $2\Delta_\ell$ of the right child of $h$. Thus,

$$\mathbb{P}(\exists h' \in T_\ell, h'' \in Subtree(h') : h'(x) \neq h''(x)) < 2^{d+1} 2\Delta_\ell.$$

Since

$$\begin{aligned} 2^{d+1} 2\Delta_\ell &= \mathbb{P}(A_\ell) \\ &= \mathbb{P}\left( x \in \bigcap_{h' \in T_{\ell+1}} \partial_{\mathcal{H}^{(\ell)}} h' \text{ and } \forall \text{ siblings } h_1, h_2 \in T_\ell, h_1(x) \neq h_2(x) \right), \end{aligned}$$

there must be some set

$$\begin{aligned} A^*_\ell = \Bigg\{ x \in \bigcap_{h' \in T_{\ell+1}} \partial_{\mathcal{H}^{(\ell)}} h' \text{ s.t. } &\forall \text{siblings } h_1, h_2 \in T_\ell, h_1(x) \neq h_2(x) \\ &\text{and } \forall h \in T_\ell, h' \in Subtree(h), h(x) = h'(x) \Bigg\} \subseteq A_\ell \end{aligned}$$

with $\mathbb{P}(A^*_\ell) > 0$. That is, for every $h$ at level $\ell + 1$, every node in its left subtree agrees with $h$ on every $x \in A^*_\ell$ and every node in its right subtree disagrees with $h$ on every $x \in A^*_\ell$. Therefore, taking any $\{x_0, x_1, x_2, \ldots, x_d\}$ such that each $x_\ell \in A^*_\ell$ creates a shatterable set (shattered by the set of leaf nodes in the tree). This contradicts VC dimension $d$, so we must have the desired claim that the maximum recursion depth is at most $d$. □

Before completing the proof of Theorem 1, we have two additional minor concerns to address. The first is that the confidence level in Lemma 1 is slightly smaller than needed for the theorem. The second is that Lemma 1 only applies when $S_p(\epsilon, \delta, h^*) < \infty$ for all $\epsilon > 0$. We can address both of these concerns with the following lemma.

**Lemma 3.** *Suppose $(C, D)$ is such that $C$ has finite VC dimension $d$, and suppose $S'_a(\epsilon, \delta, h^*)$ is a sample complexity for $(C, D)$. Then there is a sample complexity $S_a(\epsilon, \delta, h^*)$ for $(C, D)$ s.t. for any $\delta \in (0, 1/4)$ and $\epsilon \in (0, 1/2)$,*

$$S_a(\epsilon, \delta, h^*) \leq (k+2) \max \begin{cases} \min \left\{ S'_a(\epsilon/2, 4\delta, h^*), \frac{16d \log(26/\epsilon) + 8 \log(4/\delta)}{\epsilon} \right\} \\ (k+1)^2 72 \log(4(k+1)^2/\delta) \end{cases},$$

*where $k = \lceil \log(\delta/2) / \log(4\delta) \rceil$.*

*Proof.* Suppose $A'_a$ is the algorithm achieving $S'_a(\epsilon, \delta, h^*)$. Then we can define a new algorithm $A_a$ as follows. Suppose $t$ is the budget of label requests allowed of $A_a$ and $\delta$ is its confidence argument. We partition the indices of the unlabeled sequence into $k + 2$ infinite subsequences. For $i \in \{1, 2, \ldots, k\}$, let $h_i = A'_a(t/(k+2), 4\delta)$, each time running $A'_a$ on a different one of these subsequence, rather than on the full sequence.

From one of the remaining two subsequences, we request the labels of the first $t/(k+2)$ unlabeled examples and let $h_{k+1}$ denote any classifier in $C$ consistent with these labels. From the remaining subsequence, for each $i, j \in \{1, 2, \ldots, k+1\}$ s.t. $\mathbb{P}(h_i(X) \neq h_j(X)) > 0$, we find the first $\lfloor t/((k+2)(k+1)k) \rfloor$ examples $x$ s.t. $h_i(x) \neq h_j(x)$, request their labels and let $m_{ij}$ denote the number of mistakes made by $h_i$ on these labels (if $\mathbb{P}(h_i(X) \neq h_j(X)) = 0$, we let $m_{ij} = 0$). Now take as the return value of $A_a$ the classifier $h_{\hat{i}}$ where $\hat{i} = \arg\min_i \max_j m_{ij}$.

Suppose $t \geq S_a(\epsilon, \delta, h^*)$. First note that, by a Hoeffding bound argument (similar to the proof of Theorem 2), $t$ is large enough to guarantee with probability $\geq 1 - \delta/2$ that $er(h_{\hat{i}}) \leq 2 \min_i er(h_i)$. So all that remains is to show that, with probability $\geq 1 - \delta/2$, at least one of these $h_i$ has $er(h_i) \leq \epsilon/2$.

If $S'_a(\epsilon/2, 4\delta, h^*) > \frac{16d\log(26/\epsilon)+8\log(4/\delta)}{\epsilon}$, then the classic results for consistent classifiers (e.g., [20, 6, 13]) guarantee that, with probability $\geq 1-\delta/2$, $er(h_{k+1}) \leq \epsilon/2$. Otherwise, we have $t \geq (k+2)S'_a(\epsilon/2, 4\delta, h^*)$. In this case, each of $h_1, \ldots, h_k$ has an independent $\geq 1 - 4\delta$ probability of having $er(h_i) \leq \epsilon/2$. The probability at least one of them achieves this is therefore at least $1 - (4\delta)^k \geq 1 - \delta/2$. □

We are now ready to combine these lemmas to prove Theorem 1.

*Proof (Theorem 1).* Theorem 1 now follows by a simple combination of Lemmas 1 and 2, along with Theorem 2 and Lemma 3. That is, the passive learning algorithm achieving passive learning sample complexity $S_p(\epsilon, \delta, h)$ on $(C, D)$ also achieves passive sample complexity $\bar{S}_p(\epsilon, \delta, h) = \min_{\epsilon' \leq \epsilon} \lceil S_p(\epsilon', \delta, h) \rceil$ on any $(C_i, D)$, where $C_1, C_2, \ldots$ is the decomposition from Lemma 2. So Lemma 1 guarantees the existence of active learning algorithms $A_1, A_2, \ldots$ such that $A_i$ achieves a sample complexity $S_i(\epsilon, 2\delta, h) = o(\bar{S}_p(\epsilon, \delta, h))$ on $(C_i, D)$ for all $\delta > 0$ and $h \in C_i$ s.t. $\bar{S}_p(\epsilon, \delta, h)$ is finite and $\omega(1)$. Then Theorem 2 tells us that this implies the existence of an active learning algorithm based on these $A_i$ combined with Algorithm 1, achieving sample complexity $S'_a(\epsilon, 4\delta, h) = o(\bar{S}_p(\epsilon/2, \delta, h))$ on $(C, D)$, for any $\delta > 0$ and $h$ s.t. $\bar{S}_p(\epsilon/2, \delta, h)$ is always finite and is $\omega(1)$. Lemma 3 then implies the existence of an algorithm achieving sample complexity $S_a(\epsilon, \delta, h) \in O(\min\{S_a(\epsilon/2, 4\delta, h), \log(1/\epsilon)/\epsilon\}) \subseteq o(\bar{S}_p(\epsilon/4, \delta, h)) \subseteq o(S_p(\epsilon/4, \delta, h))$ for all $\delta \in (0, 1/4)$ and all $h \in C$. □

Note there is nothing special about $4$ in Theorem 1. Using a similar argument, it can be made arbitrarily close to 1.